Exploiting User Privacy in IoT Devices Using Deep Learning and its Mitigation

Rana Al Ameedee, Wonjun Lee
Department of Electrical and Computer Engineering
The University of Texas at San Antonio, Texas, USA
Email: {rana.alameedee, wonjun.lee}@utsa.edu

Abstract — **Internet of Things (IoT) has seen a great growth in recent years; the number of devices is expected to be 80 billion by 2025. Although the IoT facilitates our life, however, it threatens our privacy if we do not take the necessary security measures. In this paper, we show how the user activities can be tracked using only network traffic packets sent from several commercial IoT devices with no need for deep inspection. The prediction about daily life activities of the user at home is made based on analysis of deep learning. In addition, we propose a practical idea to mitigate the privacy attack caused by the smart home devices, and introduce experimental results showing that our approach works very accurately.**

Keywords—Internet of Thing; Smart Home; Privacy; Deep Learning.

I. INTRODUCTION

Nowadays, IoT is being used in many places where installed devices are connected to the Internet providing smart and intelligent services such as smart home, smart cities, smart health, etc. In IoT, privacy is one of the most critical terms that need to be considered due to its close connection to the life of users at home, hospital, and work. For example, a sleep-monitor device is used to track sleeping patterns, heart rate, breathing, snoring, movements of users for improving sleep quality. Sensing devices that control home objects such as light, thermometer, electricity, windows, and doors are related to human life. The research in IoT privacy has focused on keeping data hidden during its transmission to the external Internet by encrypting these data in the strong security protocols [1]. However, even though the data is hidden, by combining and analyzing data transmitted from multiple devices, a malicious party can track user's life patterns revealing critical privacy issues. Especially, smart home devices should not reveal their presence at home, because exploiting these devices with their specific function could potentially disclose personal information. Thus, even though some IoT devices in the smart home do not produce personal information, it is still possible to find out the identity of devices [2]-[4] and then, indirectly track individual's personal life style through identified devices [5].

In this paper, we show how user privacy can be exploited in deep learning model, by implementing experiments on three commercial IoT devices. Identifying the devices through the devices' manufacturer name is the first step to exploit user privacy. Training dataset for this work is generated manualy simulating human's real life pattern while testing data is extracted from the network traffic sent by devices. Analyzing in deep learning method, it was possible to show how the vulnerabilities could lead to violate user privacy by making an accurate prediction about user activities at home.

In addition to attack method, we present the most recent defense approaches and an idea that mitigates the privacy violation as well as corresponding experiment results. These results bring a broad impact on nations, as well as IoT community in the sense that human life will be based on so many different types of smart home devices that are connected to the Internet in near future. From the simple technique to get device identity information and normal traffic data with the analysis on the deep learning methods, we show that user's personal life style can be revealed. This vulnerability becomes much bigger whenever a new device is added.

The paper is composed of following four sections. Section II describes how user privacy could be exploited with deep learning; Section III presents the suitable mitigation of the vulnerabilities that are found and have caused the privacy violation; Section IV describes related and similar works to our approach including the most recent attacks and mitigations concluding in Section V.

II. EXPLOITING USER PRIVACY WITH DEEP LEARNING

In the IoT system, Domain Name Server (DNS) queries reveal IoT devices' identities since DNS queries are mapped to a specific manufacturer, when exchanging data between the devices and manufacturer's servers. The revelation of device identities alone represents privacy violations regardless of consequent attacks. For instance, some people do not want anyone to know that they use a blood pressure device or device to measure diabetes [6]. Generally, IoT devices have individual purpose with one type of data for most of time [1]. Therefore, the traffic that comes from a particular device reveals its functionality. Under such characteristics of IoT devices, identifying the devices could predict user activities in terms of functioning of devices [7].

Some IoT devices may not provide sensitive information by themselves, but when it is joined with other devices' traffic, they give strong prediction. For instance, when traffics from vacuum device, sleep sensor, and smart TV [8] are jointly analyzed together, it's possible to predict when the user goes bed to sleep [9].

The following subsections describe how we exploit user privacy by analyzing data coming from devices.

A. Smart home devices

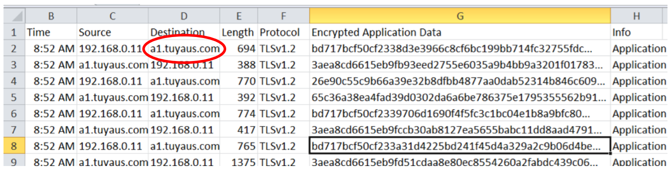
Tracking user's daily activities at home through network traffics sent by IoT devices can be performed using a deep learning method. In order to set up the IoT environment, three commercial devices are used as IoT devices; *Smart lock*, *Smart light*, *Smart alarm*. How these devices are installed in the experiment and what vulnerabilities are investigated in these devices can be found as follows:

Smart lock – Smart lock is installed on the deadbolt of the main door at home. The device sends an alarm to the user if the door is open. Device's App is installed on the phone, and the device is connected through a wireless network (i.e., Wi-Fi) to *raspberry pi3* (i.e., router in our IoT environment). From the domain name in the DNS queries which are in plain text while data transmitted between device and manufacturer's server are encrypted in Transport Layer Security (TLS), attackers can get the information about the lock and notice the identity of the device. The network traffic sent by the device denotes *open* or *close* of the door.

Smart light – Smart light is installed in the home lab and also connected to the raspberry pi router using Wi-Fi. Investigating the TLS packet header, manufacturer name (i.e., *tuyaus* in our experiment) appears clearly in the DNS queries, as shown in Figure 1. After filtering out packets other than TLS packets with the identified device name, it is found that the device sends traffics in encrypted data format whenever the device is used by the user. Thus, finding the packet here means turning *on* or *off* the light [10].

Smart alarm – Regarding device identification, DNS queries reveal device identity clearly through domain name which has a manufacturer name. Device identification through its DNS queries is a general problem for most of IoT devices including investigated six devices in [2], as well as three devices in this experiment. The smart alarm device sends encrypted data whenever it is used.

When the data coming from three devices are merged with time information and then analyzed, this analysis provides meaningful information which should not be revealed to the third persons other than users. For example, when the people have the pattern such that they wake up in the morning, turn off the light, lock the door in time order, and then no activities are sensed, it can be expected that they *left home* in the morning for a long time. If the malicious person gets a data such that the door is opened after a long time (i.e., after 16 hours), and then the light is on right after the door is open, he or she can confirm that these series of information reveal a life pattern such that the user comes back home at late night. Combining those two examples of scenario provides complete information about user's daily life such that the user wakes up in the morning, leaves home and then comes back home at late night. If the data that contains such pattern repeats multiple times, the adversary can confirm that the life pattern is really true. Based on the collected time of the same series of information, the analysis may reveal user's many different life patterns such as working at night and coming back home in the morning while sleeping at day.



	B	C	D	E	F	G	H
1	Time	Source	Destination	Length	Protocol	Encrypted Application Data	Info
2	8:52 AM	192.168.0.11	a1.tuyaus.com	694	TLSv1.2	bd717bcf50cf2338d3e3966c8cf6bc199bb714fc32755fdc...	Application
3	8:52 AM	a1.tuyaus.com	192.168.0.11	388	TLSv1.2	3aea8cd6615eb9fb93eed2755e6035a9b4bb9a3201f01783...	Application
4	8:52 AM	192.168.0.11	a1.tuyaus.com	770	TLSv1.2	26e90c55c9b66a39e32b8dfbb4877aa0dab52314b846c609...	Application
5	8:52 AM	a1.tuyaus.com	192.168.0.11	392	TLSv1.2	65c36a38ea4fad39d0302da6a6be786375e1795355562b91...	Application
6	8:52 AM	192.168.0.11	a1.tuyaus.com	774	TLSv1.2	bd717bcf50cf2339706d1690f4f5fc3c1bc04e1b8a9bfc80...	Application
7	8:52 AM	a1.tuyaus.com	192.168.0.11	417	TLSv1.2	3aea8cd6615eb9fccb30ab8127ea5655babc11dd8aad4791...	Application
8	8:52 AM	192.168.0.11	a1.tuyaus.com	765	TLSv1.2	bd717bcf50cf233a31d4225bd241f45d4a329a2c9b06d4be...	Application
9	8:52 AM	a1.tuyaus.com	192.168.0.11	1375	TLSv1.2	3aea8cd6615eb9fd51cdaa8e80ec8554260a2fabdc439c06...	Application

Figure 1. Sample raw data sent from smart light device

B. Experiment

For the experiment, we set up a home lab where raspberry pi3 is programmed as a Wi-Fi access point [11] and connected to Ethernet as an Internet provider. All IoT devices mentioned above (i.e., smart light, smart lock and smart alarm) were installed at home to be used in a real life and provided a real network packet to the simulated adversary. All the IoT devices as well as raspberry pi3 were connected to a smartphone where all IoT devices' Apps are installed to control the devices as depicted in Figure 2.

In order to get training data, network traffics are captured for a routinized 24 hours from home lab where three devices are used simulating real normal life during weekdays. If the more diverse patterns of life including weekend are collected and trained, it will give more accurate result. However, since it is good enough to show the privacy vulnerability in IoT devices even only with a weekday data, thus we used simple (e.g., 24 hours) data. The experiment consists of five steps; Capture network traffic from home lab; Filter the captured traffic; Extract features; Write training dataset; Build a deep learning model.

Capture network traffic – The entire network traffic packets coming from the router (i.e., raspberry pi) of the home lab are captured using *tcpdump* [12] and then saved as a *pcap* file.

Filter the captured traffic – First, the pcap file is opened in *Wireshark* and filtered based on DNS queries that show device identities (i.e., manufacturer name). After that, packets are separated for each device and then saved as a CSV file.

Extract features – Since the device sends encrypted data whenever it is used, sending itself gives information to the adversary about the time when users use the device. Based on the specific tasks of each device with respect to time, the adversary can predict user's living patterns by combining all

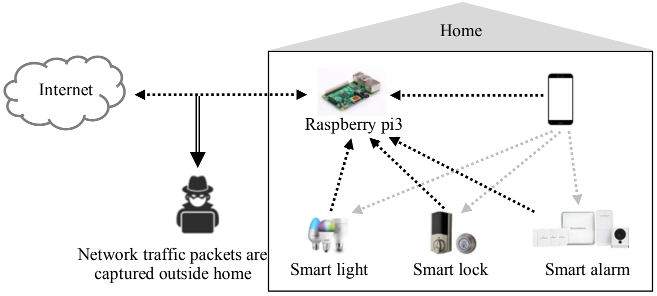
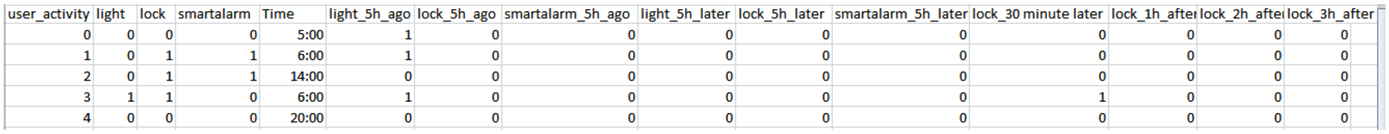


Figure 2. Experiment environment



user_activity	light	lock	smartalarm	Time	light_5h_ago	lock_5h_ago	smartalarm_5h_ago	light_5h_later	lock_5h_later	smartalarm_5h_later	lock_30 minute later	lock_1h_after	lock_2h_after	lock_3h_after
0	0	0	0	5:00	1	0	0	0	0	0	0	0	0	0
1	0	1	1	6:00	1	0	0	0	0	0	0	0	0	0
2	0	1	1	14:00	0	0	0	0	0	0	0	0	0	0
3	1	1	0	6:00	1	0	0	0	0	0	1	0	0	0
4	0	0	0	20:00	0	0	0	0	0	0	0	0	0	0

Figure 3. Samples of training data

data in CSV file to analyze. Using pandas library [13], a python code was written to extract features and merge CSV files. The features extracted for each device are as follows:

- *What time is the encrypted data sent?*
- *Which device sends this data at that time?*
- *Is this device used before five hours?*
- *Is this device used after five hours?*
- *Is the smart lock device used during the periods of 30 minutes, one hour, two hours, or three hours?*

The third feature helps the adversary to predict user activities, such as waking up or returning after a long time since users do not have any activities for five hours before waking up or returning home. The fourth feature helps the adversary to predict user activities, such as sleeping or leaving for a long time since users do not have any activities after they sleep or leave. The fifth feature helps to predict if the user left his home temporarily. All features are in binary format (i.e., *0* or *1*) except time feature which is also converted to numerical format using *sklearn* [14]. The value *1* represents that user used the device while *0* means the device was not used.

Write training dataset – The training dataset is manually written to reflect a real normal life at the same format as test dataset. This dataset considered all probabilities and times of using devices resulting in 135,668 records in CSV file. The training dataset is classified with five labels which we call, user_activity; *Wake-up*, *Leave-home-for-a-long-time*, *Return-to-home*, *Leave-home-temporarily*, and *Go-to-bed or Sleep*, which are represented as numeric values (i.e., 0, 1, 2, 3, and 4) as shown in Figure 3.

Build a deep learning **–** The model that we used is a Sequential *keras* [15] consisting of four layers including an input and output layer. All layers have 500 nodes except the last layer, which has five output nodes since we have five classes. We used a nonlinear function, *relu* as an activation function, stochastic gradient descent for optimizer, and *mean_squared_error* for loss function. After compiling the deep learning model inside a function, we used wrapper function to take *keras* model and pipe it to *scikit-learn*. In addition, *numpy* library from python was used to read and normalize data before entering it into the *keras* model.

C. *Experiment result and evaluation*

After training the model with 80% and validating with 20% of training data respectively, we obtained 98.81% of accuracy over the training dataset. Figure 4 shows that how the adversary predicted user's daily activities over 8 different scenarios in test data. The *X*-axis and *Y*-axis in Figure 4 denote predicted output and true output respectively. Our model correctly predicted user activities in that two predictions of *wake-up* for a test data were truly labeled as the same activities, and two predictions for the test data, labeled as *leaves-home-for-a-long-time* (i.e., *leave_home* in the Figure 4) were actually what they were as labeled in true outcomes. Even though there is one false result such that the model predicts *leave-home-temporarily* (i.e., *leave_temp* in the Figure 4 – up) as *leave-home-for-a-long-time*, it is still a good prediction because it does not go too far to a different class, such as *return-to-home* or *go-to-bed* predicting as a leave class. Therefore, as a result, the deep learning model provides high accuracy for the comprehensive prediction. If more IoT devices are added at home such as smart TV, smart refrigerator, smart vacuum, thermometer, camera, etc., more accurate and various personal living activities can be predicted from the accordingly added dataset. Furthermore, we can see that in the normalized confusion matrix (Figure 4 – down), the model predicted all the scenarios with accuracy of 100 %, except *leave-home-temporarily* scenario with 50%.

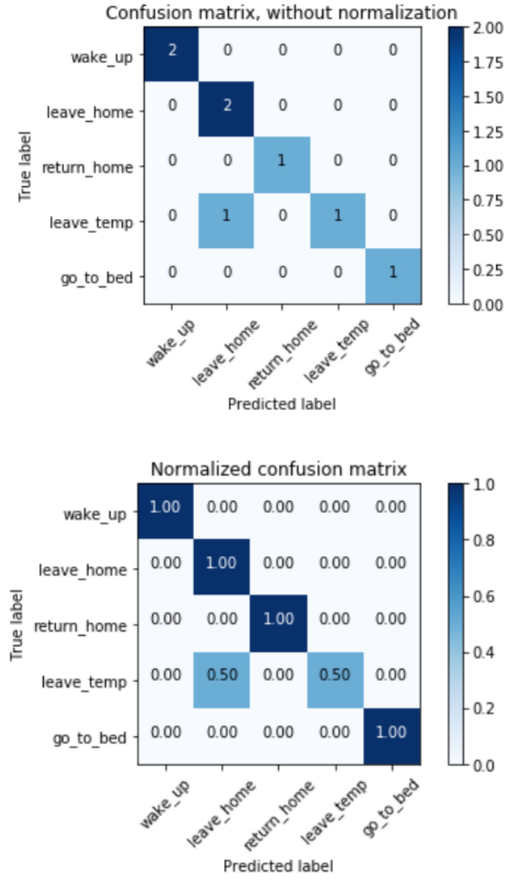


Figure 4. Up – Confusion matrix without normalization. Down – Confusion matrix with normalization.

III. Mitigation

The time of sending data from IoT devices causes the privacy violation with accurate prediction of user activities when all device traffics are merged and analyzed in deep learning model. The critical problem here is that the adversary knows when these devices are activated or used through a time of sending encrypted data. Our idea to mitigate this attack is based on the simple technique in that by sending fake data from devices, the adversary is perturbed to precisely analyze user activities. Since the data has already been encrypted, the adversary cannot distinguish fake data from the real data. We implemented the proposed idea and applied the same deep learning method to prove its effectiveness.

It is found that the prediction accuracy has been decreased to the lowest level as shown in confusion matrix of Figure 5. This figure shows how the model incorrectly predicts user activities. For example, while the adversary predicted a test data as *go-to-bed*, the actual label of that data was *wake-up*. The two test data are predicted as user's *return-to-home* but the true labels were *leave-home-for-a-long-time* (i.e., leave_home) and *leave-home-temporarily* (i.e., leave_temp). Furthermore, Figure 6 shows the regression graph between predicted and expected output describing how those two outputs match closely. *X*-axis represents testing data representing 8 scenarios, of which each is predicted to 5 corresponding output labels in *Y*-axis.

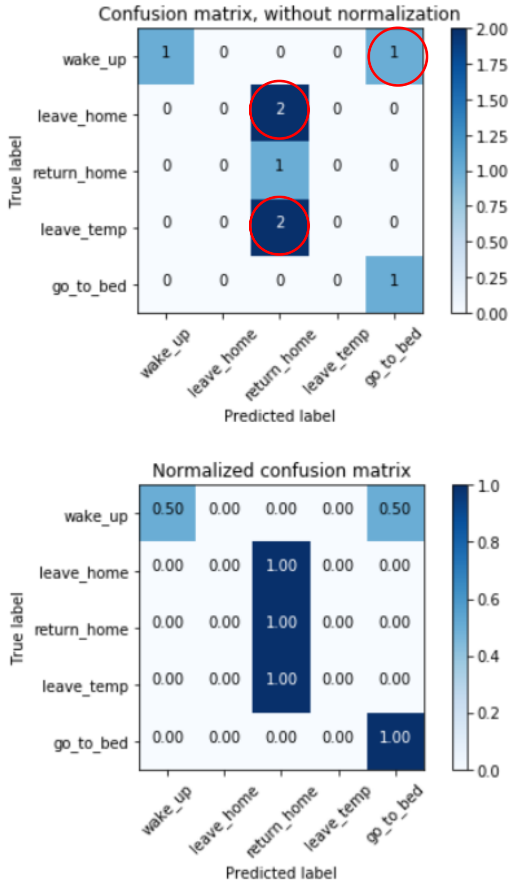


Figure 5. Confusion matrix after injecting fake data in testing data

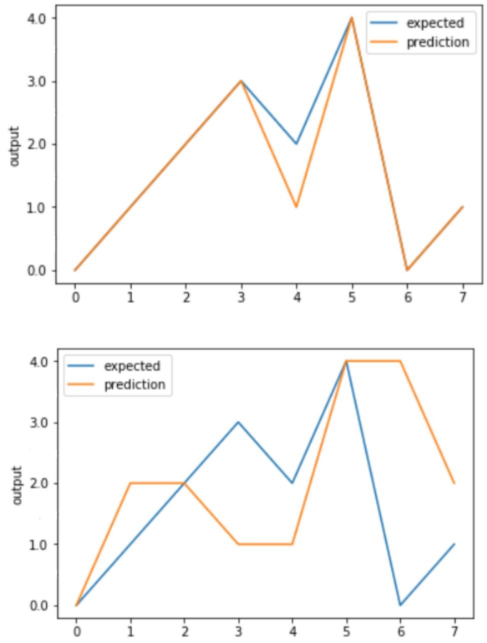


Figure 6. Regression graph without (Up) and with (Down) injecting fake data

Figure 6 – up shows that the prediction almost matches the true label meaning that the adversary successfully predicted the user activity at his home before sending fake data. Figure 6 – down shows how the prediction is far from the true labels showing that those outputs almost do not match. We can clearly see that the model predicts most of scenarios incorrectly except scenario 2 ad 5.

IV. Related Works

Apthorpe et.al. [2] analyzed four commercial IoT devices (i.e., Sense Sleep Monitor, Nest Security Camera, WeMo Switch, and Amazon Echo) to exploit user privacy. The analysis was conducted using DNS queries and metadata with no further deep inspection due to data encryption. Devices were identified through the domain name in DNS queries, which had the manufacturer name. They inferred user activities by mapping similar variations from live traffic after correlating variations of traffic rates with known user interactions. For instance, they recorded the traffic from light sensor for 12 hours at night and observed user's sleeping habits through the sent and received packet rate. After plotting this traffic, they found that traffic rate in high peak denotes user activities, such as going to bed, getting out of bed temporarily, getting out of bed in the morning. In our research, though the same approach has been used to record the traffic and identify devices using DNS queries, different IoT devices (smart lock, smart light, and smart alarm) were applied. In addition to applying to different devices, regarding network traffic packets, we only used a time of sending encrypted data and device identification information who sent it while there was no need to know the size of the sent packet, or the rate of sending these packets. In order to predict user's life style, we implemented deep learning methodologies which were only proposed by Apthorpe, et.al., [2] as their future work.

One of the proposed privacy attack mitigations is DNS concealing to prevent the adversary from recognizing the IoT device identity by Apthorpe, et.al., [4]. However, it is known that by applying a simple supervised machine learning technique, such as *k-nearest-neighbors classifier* on device traffic rates, it is possible to recognize identity of devices with accuracy 95%. Nevertheless, DNS concealing still motivates and makes device identification more complex.

A Virtual Private Network (VPN) could be used for mitigating device identification as introduced by Apthorpe, et. al. [4]. This approach includes tunneling of all traffics of a smart home to prevent the adversary from splitting the traffic into individual devices. VPN envelopes all traffic coming from the home and aggregated them into additional transport layer. Although VPN is considered a good solution to keep privacy and security for smart home, it is still possible to identify the device using supervised machine learning technique. If the home has only one IoT device, then VPN traffic rate will match the traffic from that device. In case of multiple devices, they send traffic at a different time in that the adversary still could identify the devices. For instance, a smart door lock and smart sleep monitor are less likely to be recording user activities simultaneously because it is impossible for the user to sleep and open the door at the same time.

One of the proposed approaches to mitigate attacks that use revelation of device usage patterns is to make IoT devices delay sending data to the server. For instance, sleep sensing device delays sending data for a couple of hours instead of sending right away to server. This mitigation could be successful for devices, such as sleep sensors that do not require direct outcome; however, devices that require a real-time response to triggers, such as smart alarm or personal assistant devices cannot be used for this type of mitigation because those devices cannot wait to answer user's question [4].

V. Conclusion

Smart home devices connected to Internet provide not only convenience in life to human but also private information of users to adversary. Even though many smart devices are prevalently being used in many places including house, many people are not aware of vulnerabilities that reveal their life pattern and its potential threats of misuse by malicious parties.

In this paper, we presented a deep learning based privacy attack method and its mitigation in IoT environment. From the experiment, we showed that data encryption is not enough to assure user privacy when using smart home devices and it requires additional technique to hide device usage patterns represented in time by adding noisy traffics.

As a future work, we will add additional mitigation methods such as sending fake traffics in random period, as well as its performance analysis.

References

[1] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported Internet of Things", IEEE Internet of Things Journal, vol. 3, no. 3, pp. 269-284, 2016.

[2] N. Apthorpe, D. Reisman, and N. Feamster, "A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic", 2017, *arXiv preprint arXiv:1705.06805.*

[3] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis", In Proceedings of the Symposium on Applied Computing, pp. 506-509, ACM, April 2017.

[4] N. Apthorpe, D. Reisman, and N. Feamster, "Closing the Blinds: Four Strategies for Protecting Smart Home Privacy from Network Observers", 2017, arXiv preprint arXiv:1705.06809.

[5] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini, "Security and privacy issues for an IoT based smart home. In Information and Communication Technology", The 40th International Convention on Electronics and Microelectronics (MIPRO), pp. 1292-1297, IEEE, May 2017.

[6] C. Debes et al., "Monitoring activities of daily living in smart homes: Understanding human be-havior", IEEE Signal Processing Magazine, vol. 33, no. 2, pp. 81-94, 2016.

[7] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments", Information, vol. 7, no. 3, 44, 2016;7:44 doi: 10.3390/info7030044.

[8] R. L. Rutledge, A. K. Massey, and A. I. Antón, "Privacy impacts of IoT devices: a SmartTV case study", IEEE International Conference in Requirements Engineering Conference Workshops, pp. 261-270, September 2016.

[9] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: State of the art and technical challenges", Pervasive and Mobile Computing, 17, pp. 159-174, 2015.

[10] J. Huntley, "DoctorBeet's Blog: LG Smart TVs logging USB filenames and viewing info to LG servers": http://doctorbeet.blogspot.com/2013/11/lg-smart-tvs-logging-usb-filenames-and.html [retrieved: 07, 2018].

[11] L. Ada and T. Adafruit, "Setting up a Raspberry Pi as a WiFi access point", nd): n. pag. Adafruit. Adafruit, 5, 2016.

[12] N. Kumar, J. Madhuri, and M. ChanneGowda, "Review on security and privacy concerns in Internet of things", International Conference on IoT and Application (ICIoT), pp. 1-5, IEEE, May 2017.

[13] W. McKinney et al., "Pandas: Python Data Analysis Library": https://pandas.pydata.org/ [retrieved: 07, 2018].

[14] scikit-learn developers, "scikit-learn": http://scikit-learn.org/stable/index.html [retrieved: 07, 2018].

[15] Keras Google group, "Keras: The Python Deep Learning library": https://keras.io/ [retrieved: 07, 2018].