Towards a Smart Identification of resource constrained IoT Devices Over the Air

Jared Lynskey, Dr. CS Hong
jared@khu.ac.kr, cshong@khu.ac.kr
Kyung Hee University, South Korea

Abstract

IoT has brought unprecedented changes in the world communication networks, including everyday devices that were not considered internet capable are continuing to explode in numbers. However, device management is a problem that needs to be addressed due to the large population of devices, especially in urban areas with a large number of units transmitting. In this paper we propose a new solution applying an ensemble of machine learning methods that can correctly identify a device with an accuracy of 98%, outperforming the results of singular machine learning classifiers. With these promising results our goal is to apply it to real-world applications such as device management and network security.

1. Introduction

IoT has brought unprecedented impact to the internet as we know, with devices continuing to be connected the internet including everyday objects that were not considered to be internet capable. One dilemma that arises in areas with a considerably large number of devices transmitting over the air is identifying the source including device and model type based on only network traffic. For example, it will be possible for network administrators that manage a large number of IoT devices to apply this solution in order to identify current devices transmitting to a gate way, with IoT security [1] finding that 71% of IoT are not monitored real-time.

2. Previous Work

Meidan [2] proposes a packet forwarding system security by analyzing data on the network and comparing the output from the classification model against a whitelist of known TCP/IP devices. E Hodo [3] presents a threat analysis of the IoT and uses an Artificial Neural Network (ANN) to combat these threats. Our paper is the first paper to create a device type classification for UDP/IP which is likely to be one of the widely used protocol for IoT resourced constrained devices due to its smaller overhead compared with TCP/IP.

3 Data collection Process

Resource constrained devices are typically found at the edge within a 100 meters of an gateway, therefore the dataset used in this paper were gathered directly from the device with a wireless card packets sent over the air from four devices. Furthermore, to ensure that our dataset was suffice to make classifications, devices including the smart door lock sensor and smart light sensor, were both purposely used in order to ensure packets were captured from the two devices. Other devices were able to produce enough packets in order for our model to generate accurate results under normal network conditions.

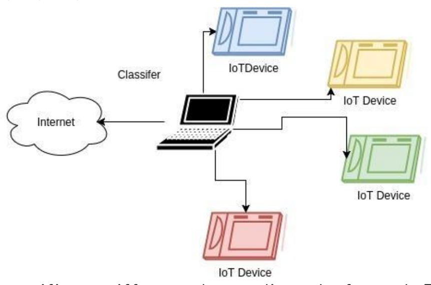


Figure 1 Classifier sniffs packets directly from IoT devices

4. Devices and Feature dataset characteristics

The following table contains the resource constrained IoT devices monitored, packets captured and number of UDP sessions.

Device	Total number of UDP Packets captured	Total number of UDP sessions
Air quality Sensor	3842	144
Motion sensor	6234	340
Smart Door Lock	700	24
Smart Light Switch	1123	48

Figure 2. device dataset contents

The following features were chosen due the fact that the majority of information can be found encapsulated in the UDP packet

Features of sample in dataset
Source IP
Destination IP
Application protocol (e.g CoAP)
Original size of Data
Trimmed data content*
Device name

Figure 3. device features captured

5. Proposed classification and preprocessing

Firstly, due to the range of wide range of values captured during packet sniffing, it is necessary to standardize to fit the data between a range in order to reduce the variance. Secondly, non-discrete values are unsuitable as input for classification algorithms and must be encoded into numerical values, his applies to the features such as source IP and destination IP. Thirdly, to further optimize the learning process, we considered the 'payload size', minimizing this field would greatly reduce the dataset size and processing time of our ensemble. X is the sample data, P is the set of tested payload lengths, C is the set of classifiers, ŷ is the predicted device, acc is the accuracy score, p* is the chosen payload length from the maximum accuracy mean value from the classifiers.

```
Algorithm 1: Find optimal Payload Length
Result: $p^*$
initialization $\mathbf{C}, \mathbf{P}, \mathbf{X}$;
for each $p \in \mathbf{P}$ do
    for each $c \in \mathbf{C}$ do
        $\hat{y}^{(i)} \leftarrow$ Classify $(x_p^{(i)})$;
        $acc_c \leftarrow$ Score $(y^{(i)}, \hat{y}^{(i)})$;
    end
    $acc_p \leftarrow mean(acc_c)$;
end
$p^* \leftarrow \max\ (acc_p)$
```

Figure 4. optimizing payload process

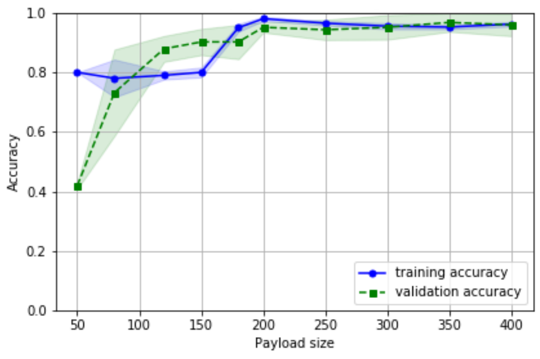


Figure 5. payload size accuracy curve

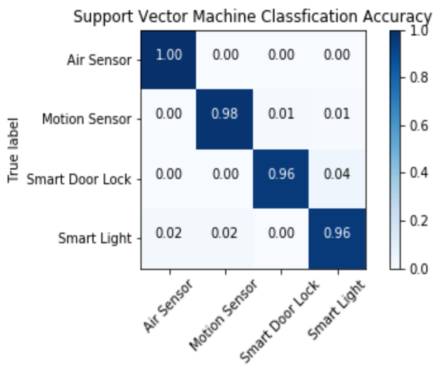
From the training results in figure 5, a payload size of approx 200kb is the most effective size in order to get the most effective classification accuracy from the payload. Lastly, To generate a model that reduces error caused by individual machine learning methods we use an ensemble [4]. The three machine learning algorithms in the ensemble are denoted as C: Logistic Regression, SVM and KNeighborsClassifier.

```
Algorithm 2: Ensemble Classification Method
Result: accuracy
initialization $\mathbf{X}, \mathbf{C}$;
for $x \in \mathbf{X}$ do
    for $x \in \mathbf{C}$ do
        $y_c^{(i)} \leftarrow$ Classify$(x^{(i)})$;
    end
    $\hat{y} \leftarrow$ mode $\{y_c^{(i)}, y_{c+1}^{(i)}, ..., y_{c+n}^{(i)}, ...\}$;
end
accuracy$\leftarrow$ Score $(\hat{y}, y)$
```

Figure 6. Ensemble method

6. Machine Learning Classification Results

To illustrate the advantage of ensemble over a single classification model we have the normalized accuracy of classifying the devices with each single classification algorithms in figures 7, 8 and 9.



True label	Air Sensor	Motion Sensor	Smart Door Lock	Smart Light
Air Sensor	1.00	0.00	0.00	0.00
Motion Sensor	0.00	0.98	0.01	0.01
Smart Door Lock	0.00	0.00	0.96	0.04
Smart Light	0.02	0.02	0.00	0.96

Figure 7. SVM classification accuracy

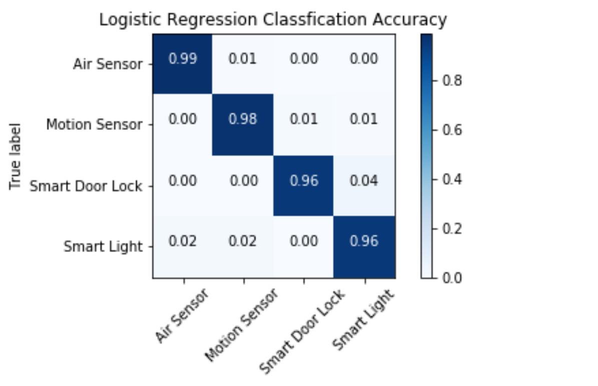


Figure 8. Logistic Regression classification accuracy

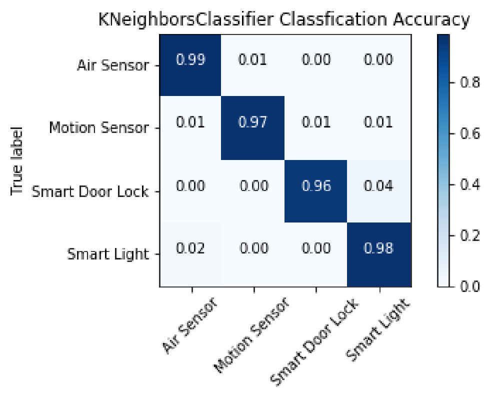


Figure 9. Neighbors Classification Accuracy

Since the ensemble takes the majority output, there is a less likely chance that the incorrect classification output is chosen, reducing the overall error. Figure 10 below shows the normalized classification accuracy for each device. The combination of the classification algorithms has improved the accuracy by up to 2% in some cases.

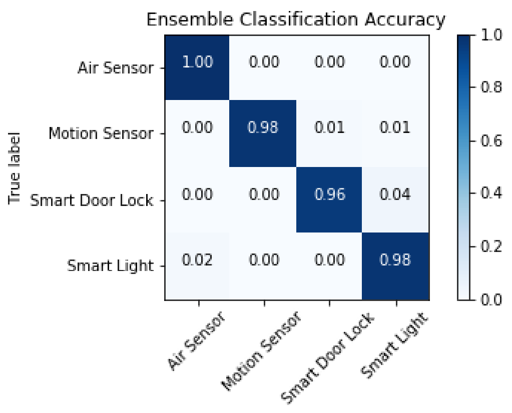


Figure 10. Ensemble Classification Accuracy

7. Conclusion

In our paper, we propose and apply a new machine learning process to IoT resource constrained device to classify the device type. Furthermore, our work findings found that the entire payload content is not necessary when training a machine learning model. The results show the proposed method outperforms methods that are used to classify devices based on TCP/IP traffic.

8. Future Work

With these promising results, our future plan is to create an online learning method to collect packets and classify traffic on the go for IoT access control and intrusion detection.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00567, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks)
*Dr. CS Hong is the corresponding author

References

[1] IoT security: the majority of IoT devices is not monitored in real time [Online]
Available: https://www.i-scoop.eu/iot-security-majority-iot-devices-not-monitored-real-time/
27 October 2017
[2] Yair Meidan (2017), Detection of Unauthorized IoT Devices Using Mto traditional methods achine Learning Techniques. Available:
https://arxiv.org/pdf/1709.04647.pdf
[3] E Hodo (2017) Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System. Available:
https://arxiv.org/pdf/1704.02286.pdf
[4] Tansu Alpcan (2010) Ensemble Methods in Machine Learning. Available:
http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf
[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.