Bluetooth Operation Classification to Detect the Anomaly Behavior by Monitoring the Power Amplifier Signal

Abdelrahman Elkanishy

Motivation

Bluetooth is considered the most common choice for wireless communication in compact devices such as smartwatches, headsets, fitness trackers, and smartphones. In addition, through its extensive usage in Internet of Things (IoT) devices, Bluetooth has developed over the past 20 years to provide low energy consumption and high transfer rates which fit perfectly for area- and power-limited devices [1]. Due to the complexity of the integrated smart devices that use the Bluetooth communication protocol, most manufacturers are outsourcing the chips to third-party suppliers. The integration between many outsourced chips has resulted in the need to add a hardware security layer to ensure appropriate operation. For example, in Apple's smartphones, there is a dedicated a co-processor, Secure Enclave, to handle all cryptographic operations and maintain the integrity of data protection [2].

Manufacturers normally buy their Bluetooth chips from third-party suppliers, which are then integrated into a complex hardware-software stack with a variety of potential vulnerabilities. Direct measurement of the output and input power signals can help security functions prevent unauthorized data transmission. Bluetooth, as any communication protocol, has vulnerabilities. On the other hand, researchers disclose those attacks in order to provide security updates to defend users' devices. For instance, in 2017, Armis identified a new Bluetooth attack vector called BlueBorne that can take control of the target device. BlueBorne attacks regular computers, smartphones, and IoT devices [3]. This security breach occurs without pairing to the targeted device nor even while the Bluetooth is in discovery mode. As the Bluetooth chip is responsible for establishing the connection and controlling the flow of data, BlueBorne and other security breaches could attack the Bluetooth communication chip without consent of the controller chip. Therefore, monitoring a Bluetooth chip at the hardware level is necessary to verify its correct operation.

This project is part of bigger project which proposes a compact supervisory circuit to classify the operation of a Bluetooth chip at low frequencies by monitoring the radio frequency (RF) output and the power input lines of the Bluetooth chip. The idea is that classification algorithm can be inexpensively fabricated on a low frequency integrated circuit in legacy technology and/or with minimal area. When the supervisory circuit detects abnormal behavior, it can be configured to shut down the Bluetooth chip. Using features extracted from the power lines, we are able to train several machine learning (ML) algorithms to classify different Bluetooth operation modes and parameters such as operation profile, distance between the paired devices, and number of connected devices. Previously, we already investigated the RF output signals and trained ML models that can separate Bluetooth advertising and transmit/receive modes with ~ 100% accuracy and classify the operation profile of the Bluetooth chip with ~ 100% accuracy.

Problem

One way to monitor the chip is to consider it as a black box which consumes and transmits power. Thus, abnormal behavior can be detected by learning the normal input/output (I/O) power signatures or by parameterizing aspects of the Bluetooth connection (e.g., profile type, distance between paired devices, and number of connected devices) and comparing to the controller chip instructions. For this project, I am aiming to inspect the power input of the power amplifier which drives the chip's antenna. Also, I will investigate can the Power Amplifier signal replace the RF signal to get the same classification accuracy. In this project, I will investigate another power signal domain of the Bluetooth chip which is the IVDD (Current Voltage Drain-Drain) of the power amplifier that drives the antenna. This power domain will be better for the electronic circuit implementation as it is not a RF signal. The measurement of RF signals requires to double the RF power in order to compensate the effect of power splitting. On the other hand, the IVDD measurement overhead only 1% of the power amplifier signal.

Solution

Collecting the IVDD signal stream. Next, I will utilize MATLAB to extract features from the CSV files of the signals which produced by the oscilloscope. Then, train models to classify the profile and the mode of transmission.

Lab Setup

We prototype our supervisory circuit using evaluation boards and an oscilloscope in order to collect a data set sufficient for training and testing. An evaluation board for the CYW20706 Bluetooth chip is used to program and emulate different profiles and events. The CYW920706WCDEVAL evaluation board supports pinouts for measuring the power lines of the CYW20706 Bluetooth chip. Also, it allows a USB connection with a computer for programming the chip and controlling it while running the program. The Bluetooth board is programmed to act as 2 popular profiles: hands-free and headset, in addition to customized profiles using GATT services. While each profile is running, different events are occurring (dialing, hang-up, streaming music, etc.). The events are controlled using a graphical user interface on a laptop.

Project Tasks

1- Collecting the envelope of the RF signal and power amplifier IVDD signal at the same time using the oscilloscope at a sampling rate not more than 1MHz which is the sampling frequency of the ADC on the FPGA board (DE10-Lite Board). The project is limited with this frequency rate which maybe results an aliased sampled signal. Thus, one of the biggest challenges of this project is to relay on features of the aliased signal.

2- Using MATLAB, I will extract features from the power streams that can be denote to the operation of the Bluetooth chip. The project is focused on relatively big processing window of 640ms which can include huge number of transmission events. Consequently, the aim of the algorithm is to track the overall behavior of the chip.

3- Training different Machine learning algorithms such as, such as decision tree, K-Nearest Neighbor (KNN), support vector machine (SVM), and quadratic discriminant analysis, for the purpose of comparing their accuracy and prediction speed. For all classifiers, I will use cross validation for testing the models. Then, I will choose the best algorithm in accuracy, prediction speed, and the lowest computational implementation. The training will done using the classification learner tool of MATLAB.

4- Comparing the performance of the power amplifier classifiers and the enveloped RF ones from the points of accuracy, predication speed, and the computational complexity. In order to hardware implementation of the pre-trained model.

Tasks Progress: (10/17/18)

1- Data Collection:

The Bluetooth board was programmed with different events and profile, then the RF and Power Amplifier Signal were captured using the oscilloscope which saves the signals in a CSV file format. The Bluetooth board is programmed to act as 2 popular profiles: hands-free and headset, in addition to customized profiles using GATT services. While each profile is running, different events are occurring (dialing, hangup, streaming music, etc.). The events are controlled using a graphical user interface which utilizes a serial port through USB to send commands to the Bluetooth evaluation board. A network topology of two devices is defined. Moreover, we collect the RF streams of each profile in both the advertising and transmitting/receiving (trans-receiving) states. First, the hands-free profile RF output signal is recorded while executing multiple events, such as dialing, answering, and hang-up. Second, the headset profile RF output signal is captured during various events such as streaming music, rewind, scrub, and volume control. Lastly, a customized profile is used to simulate a simple embedded system which can be connected through Bluetooth communication. Basically, it notifies the Bluetooth evaluation board of a sensor reading to a paired device, which can control the number of blinks of a light-emitting diode.

2- Data Preprocessing and Extracting Features:

Based on investigating the collected signals, three features are extracted from each window to train the ML models. The first feature extracted is the maximum signal value in the given window, since the maximum signal value is expected to vary from one transmitting state to another. Fluctuations in the maximum value are related with the different profiles. As we are interested in the pattern of the Bluetooth transmission, the other two features are extracted after thresholding the envelope-detected stream into two binary levels. In other words, the signal is 1-bit quantized. Therefore, after quantization, value 1 means the Bluetooth is transmitting, whereas value 0 indicates no transmission. The remaining two features extracted are the area under the curve and the number pulses, both extracted from the 1-bit quantized signal in a given window. The area under the curve is correlated to the total transmission duration in a certain window, whereas the number of pulses represents the density of the transmission events in the window.

I wrote a MATLAB script, named “Features_Extraction_LDO”, to pre-process the data. Then, the features are extracted. Next, the data is formulated into table format which is compatible with the MATLAB Classification Learner App. I run the script on each captured signal CSV file while changing “profile” and “Mode” parameters for each file.

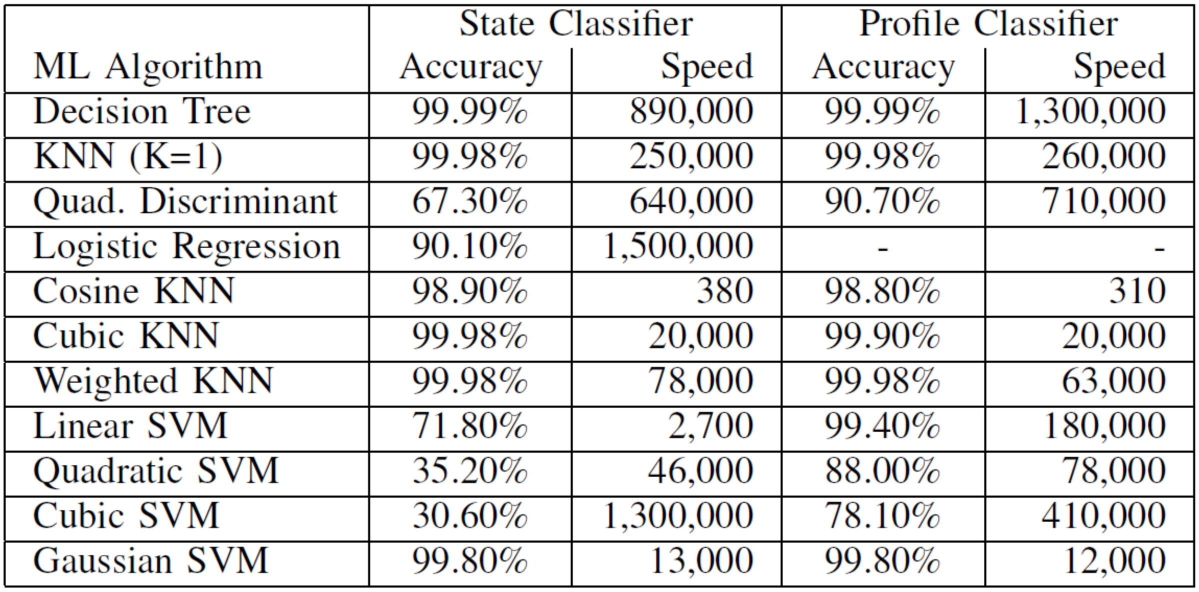
Script Description

There are two MATLAB scripts to pre-process the data and extract the features. The two scripts are named as "Features_Extraction_LDO.m" for the power amplifier signals and "Features_Extraction_RF.m" for the Radio Frequency (RF) signals. "Plot_Raw_Data.m" script is used to visualize the raw data in plots. The two feature extraction script were ran on the raw data which there is an example of it in the datasets folder. The example of raw data file is named “Raw_Sensor_Data_Advertising.csv". The features extraction scripts commented in way that can change the generation of the data set classes such as "profile" and "mode" depending on the raw data file.

Dataset Description

In the datasets folder, there are two ".mat" files which contain the data sets were used to train and test the models. "HS_HF_S_LDO.mat" is for the power amplifier signals while "HS_HF_S_RF.mat" is for RF signals. Each dataset includes 3 attributes. Each instant is labeled by its’ the mode and the profile. LDO dataset is 298, 293 instances while RF dataset is 274, 681 instances. The three features are the peak of the window and area under the curve and number of pulses in a given window. The profile classes are headset, hands-free, and sensor. While, the mode classes are Advertising and trans-receiving.

Results and Discussion



ML Algorithm	State Classifier Accuracy	State Classifier Speed	Profile Classifier Accuracy	Profile Classifier Speed
Decision Tree	99.99%	890,000	99.99%	1,300,000
KNN (K=1)	99.98%	250,000	99.98%	260,000
Quad. Discriminant	67.30%	640,000	90.70%	710,000
Logistic Regression	90.10%	1,500,000	-	-
Cosine KNN	98.90%	380	98.80%	310
Cubic KNN	99.98%	20,000	99.90%	20,000
Weighted KNN	99.98%	78,000	99.98%	63,000
Linear SVM	71.80%	2,700	99.40%	180,000
Quadratic SVM	35.20%	46,000	88.00%	78,000
Cubic SVM	30.60%	1,300,000	78.10%	410,000
Gaussian SVM	99.80%	13,000	99.80%	12,000

Table 1 COMPARING DIFFERENT MACHINE LEARNING (ML) MODELS IN TERMS OF ACCURACY IN PERCENTAGE (%) AND PREDICTION SPEED (PS), MEASURED IN OBSERVATIONS PER SECOND

Table 1 presents performance for various ML algorithms which are applied to classify the transmission state (State Classifier) and the profile (Profile Classifier). We present the prediction speed after training and the classification accuracy of different ML algorithms. As can be seen, for the State Classifier, logistic regression is the fastest algorithm in prediction, but is less accurate than other 6 tested algorithms. Among the remaining 6 algorithms, the decision tree is the most accurate model with 99:99% accuracy and also the second fastest predictor. The KNN algorithm has a high accuracy with an average 99:7% over the different flavors of KNN (1-NN, cosine, cubic, and weighted). Of those, KNN with k=1 (1-NN) and weighted KNN have the highest prediction accuracy out of the varieties of KNN and 1-NN is the fastest in prediction. Looking simultaneously for both high accuracy and prediction speed, 1-NN and decision tree are the top candidates for the State Classifier. Nevertheless, decision tree is 3.5 times faster than 1-NN in prediction. Also, regarding the hardware implementation of the models, the 1-NN model needs more storage than that of the decision tree, because 1-NN keeps a copy of the training data in order to calculate the Euclidean distance between the prediction point and the nearest training set observation. The point is then classified according to the class of the closest observation. In contrast, the implementation of the decision tree algorithms is based on branching, with a maximum number of branches per feature of 100 in this case. Therefore, the computational load of decision tree is much less than that of 1-NN. For the Profile Classifier, the decision tree surpasses all other algorithms in both accuracy and prediction speed. Despite only a small difference in accuracy between decision tree and 1-NN, the prediction speed of decision tree is 5 faster than 1-NN. Similar to the decision tree State Classifier, the maximum number of branches in the decision tree Profile Classifier is set to 100. Hence, the decision tree again has lower computational load than 1-KNN, and there is a large difference in the prediction speed between them.

	RF Output	Power Amplifier
Observations	274, 681	298, 293
Decision Tree (PC)	99.99%	99.99%
KNN (PC)	99.98%	99.98%
Decision Tree (SC)	99.99%	99.99%
KNN (SC)	99.98%	99.99%

Table 2 Comparison between the power amplifier and RF results.

Table 2 shows the differences between applying the machine learning algorithms on two different signals of the Bluetooth chip. For the same events, the power amplifier's unique observations are more than that of the RF. Consequently, the power amplifier domain has more information than the RF domain. The accuracies of the KNN and decision tree algorithms on both signals are so close to each other. Also, one of the two domains are enough to have a high classification accuracy.

Conclusion

- The mode and profile classifiers enable the devices that use Bluetooth to keep track of the physical layer with ~100% accuracy using K fold cross validation (K=5).
- The selected three simple features which are extracted from the chip RF output are enough to achieve high classification accuracy with 4 bit depth.
- Power Amplifier and RF signals are highly correlated.
- Power Amplifier Signal has slightly more information.

References

[1] K. Chang, "Bluetooth: a viable solution for iot? [industry perspectives]," IEEE Wireless Communications, vol. 21, no. 6, pp. 6–7, December 2014.

[2] iOS Security, iOS 11, January 2018. [Online]. Available: https://www.apple.com/business/docs/iOS Security Guide.pdf

[3] The Attack Vector BlueBorne Exposes Almost Every Connected Device. [Online]. Available: https://www.armis.com/blueborne/