# Physical Fingerprinting of Ultrasonic Sensors and Applications to Sensor Security

Eric Cheek
*School of Electrical and Computer Engineering*
*University of Michigan - Dearborn*
*Dearborn, Michigan USA*
*Email: echeek@umich.edu*

Dhimant Khuttan
*School of Electrical and Computer Engineering*
*University of Michigan - Dearborn*
*Dearborn, Michigan USA*
*Email: dkhuttan@umich.edu*

Raghu Changalvala
*School of Electrical and Computer Engineering*
*University of Michigan - Dearborn*
*Dearborn, Michigan USA*
*Email: rchangal@umich.edu*

Dr. Hafiz Malik
*School of Electrical and Computer Engineering*
*University of Michigan - Dearborn*
*Dearborn, Michigan USA*
*Email: hafiz@umich.edu*

***Abstract*—As the market for autonomous vehicles advances, a need for robust safety protocols also increases. Autonomous vehicles rely on sensors to understand their operating environment. Active sensors such as camera, LiDAR, ultrasonic, and radar are vulnerable to physical channel attacks. One way to counter these attacks is to pattern match the sensor data with its own unique physical distortions, commonly referred to as a fingerprint. This fingerprint exists because of how the sensor was manufactured, and it can be used to determine the transmitting sensor from the received waveform. In this paper, using an ultrasonic sensor, we establish that there exists a specific distortion profile in the transmitted waveform called physical fingerprint that can be attributed to their intrinsic characteristics. We propose a joint time-frequency analysis-based framework for ultrasonic sensor fingerprint extraction and use it as a feature to train a Naive Bayes classifier. The trained model is used for transmitter identification from the received physical waveform.**

## 1. Introduction

Active sensors [1] such as ultrasonic sensors and many others have been employed within autonomous vehicles in order to perceive and localize themselves within their surrounding environment. Within a vehicle, all information, including sensory data for localization, is passed between Electronic Control Units (ECUs) over the Controller Area Network (CAN). These ECUs typically control an electrical subsystem within a vehicle, such as the Powertrain Control Module (PCM), Transmission Control Module (TCM), Brake Control Module (BCM) and so on.

Once data arrives at an ECU over the CAN network, that particular ECU utilizes the data to control the subsystem it pertains to and ultimately allows for the vehicle to operate. As is, there is no validation of data transmitted over the CAN network [2] or between ECUs and microcontroller units (MCUs) within an Advanced Driver Assistance System (ADAS) [3]. There is also not known to be any security measures being taken on the sensor level before data is sent to various ECUs or aforementioned CAN and ADAS systems. The absence of a secure protocol for recording sensor data and transferring data between ECUs in a vehicle creates a hesitance towards pushing autonomous vehicles to market. In an autonomous vehicle, corrupted data or false sensor readings could be the difference between life or death.

This paper focuses on extracting sensor intrinsic properties called **fingerprints** that can serve as a potential countermeasure for two physical signal level attacks, which are attacks categorized by manipulating the environment in such a way to cause incorrect ultrasonic sensor measurements.

Fingerprints are formed due to microscopic imperfections and dissimilarities in the sensor manufacturing process. They are physical features prevalent in a multitude of Cyber-Physical Systems (CPS) and other hardware devices that arise in specific waveform characteristics. Sensor fingerprints can be represented as a function of the material properties which make up a sensor or a piece of hardware and fabrication process. These imperfections are assumed to be unique to a specific sensor and random in nature. The concept of physical fingerprinting has been used for RF transmitter identification [4] and hardware validation for sensors used in non-automotive applications [5]. In case of ultrasonic sensors, this fingerprint manifests in the form of random noise in the transmitted pulse sequence from the sensor which can be observed in the sensor transmissions.

### 1.1. Motivation

The motivation for this work lies in the innovation of self-driving vehicles that utilize various sensors to perceive

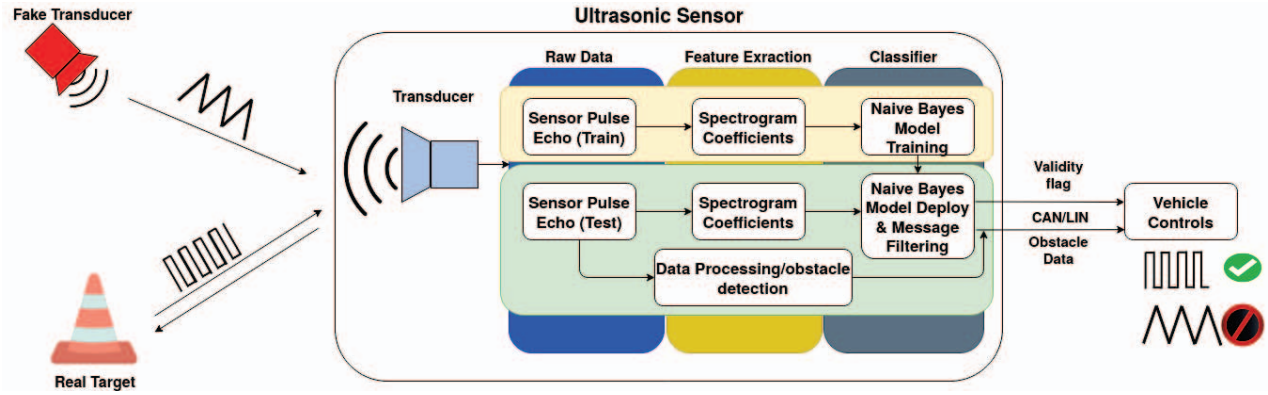


Figure 1: Block-diagram of the system model

their environment. For this technology to become more prevalent, it must be safe and robust to malicious attacks. Ultrasonic sensors are already common in modern vehicles for close-range object detection, such as in the Tesla Model S, which relies on ultrasonic sensors to achieve its "Smart Summon" feature. It has been successfully demonstrated in [6] that these sensors are vulnerable to attack and can have fatal outcomes. Other modern vehicles rely on ultrasonic sensors to assist drivers when parking. When an ultrasonic sensor is attacked employing jamming and spoofing, it can perceive an object that is not truly there (false positive), or cause the sensor not to perceive an object that truly is there (false negative) as shown in [7]. These incorrect readings can cause damage to a vehicle, building, and human life. This work serves as one of the initial steps in securing the technology used in autonomous vehicles, so it becomes resistant to attacks.

### 1.2. Principal Contribution

Our method of fingerprint generation and implementation is novel in the amount of data that is required to establish a reliable fingerprint, the accuracy of the classifier due to our decision for identifying fingerprints, distance invariability of our fingerprint as well as what features we take into consideration for our classifier. In theory, it is mathematically impossible to spoof a signal with the same random noise pattern. The other proposed methods are theoretically more vulnerable to attacks because a knowledgeable attacker can listen and craft ultrasonic pulses with similar amplitude, resonance frequency and phase. It does not rely on time-domain statistical measures of the transmitted waveform as in [7] and [5]. Our method also does not require any rigorous modification to an existing sensor system or require us to handcraft our own pulses as in [7]. Our work differs from [5] and [7] in that our system performs with equivalent or better accuracy (Tab 1) when identifying sensors via a significantly smaller feature vector. Our feature extraction algorithm uses adaptive filtering to remove irrelevant frequencies and only generates feature vectors for a specific bandwidth which is also novel and attributes to the accuracy of our classifier. Our algorithm computes the fingerprint of a given sensor by analyzing the energy component around the resonance frequency ( 42kHz in our case) where the energy is highest.

We propose a method to extract the sensor fingerprints by observing the spectrogram for each sensor at multiple distances to determine each sensor's respective resonance frequency (Fig 3). Once we have determined the resonance frequency for a sensor, our algorithm extracts data from the necessary frequencies which will create a frequency profile used for training our classifier. By applying a band-pass filter to our data, our classifier ignores irrelevant data and is in turn more accurate. Each frequency bin in our spectrogram over the desired interval acts as a feature vector later for our classifier and essentially contains the fingerprinting information of a given sensor. As mentioned, pattern matching a fingerprint to a specific sensor based on spectral content is extremely robust, since it is infeasible for an attacker to generate and transmit an ultrasonic waveform with the same fingerprint or random noise profile, even if the attacker has sophisticated knowledge of our implementation. In the case an attack was to occur, our classification method can be extended using a Support Vector Machine (SVM) to classify an input that does not belong to any classes the model is trained on. However, this is not implemented in this paper. We train a simple, computationally light machine learning model with this feature to demonstrate that the transmitting sensor can be identified through physical fingerprint.

## 2. Related Work

Recently, research efforts have been focused on leveraging physical signal features to detect malicious attacks [5], [7], [8]. In [7], single-sensor-based physical shift authentication and multiple-sensor consistency checks were employed to verify signals on a system level. In the first method of countering sensor attacks, physical aspects of the transmitted waveform were altered, such as amplitude, frequency and phase. At the receiver, a correlation function is applied to the received echo and transmitted signal. If the correlation

value exceeded a threshold, then the received waveform is accepted by the system. Multiple-sensor consistency checks utilize the time of flight (TOF) of each received pulse and compare them to each other. If a discrepancy exists which exceeds the tolerance, then an attack most likely occurred. If an attack occurred but the TOF for each echo are very closely related, then the attack is not a threat to the system. This method for attack detection is different from ours, since we use a machine learning approach where no manipulation of the waveform characteristics is necessary to detect sensor attacks.

In [5], researchers again worked to find ways of countering ultrasonic sensor attack. Data was collected for ultrasonic sensors on stationary objects. Data chunks were formed and both time domain and frequency domain features were used to determine whether or not an attack had occurred. The features used in order to detect attacks were the mean, standard deviation, mean average deviation and kurtosis of the time domain signal as well as spectral standard deviation, spectral centroid and DC component of the spectral signal. Hardware fingerprints were shown to exist in a few lower-cost HC-SR04 ultrasonic sensors using the aforementioned features. Our work wishes to expound on this by using more expensive ultrasonic sensors which are expected to have less variance due to manufacturing processes and higher-grade materials. Also, our work wishes to be able to identify from what sensor and at what specific distance an ultrasonic waveform traveled simply by feature extraction and classification.

Work has also been done in [8] to prevent the vulnerability of ultrasonic sensors to malicious attacks by using hardware fingerprints in water level monitoring applications. Noise patterns were created by the process of the filling of water containment vessels, measuring a distance with the ultrasonic sensor and comparing to the ground truth values (the flow rate was known). After several runs were recorded, a pattern was established. If the noise pattern of an ultrasonic sensor differed from the amount expected at a specific fill rate, then an attack was detected. This method is robust but differs from ours since we are proposing a method for hardware fingerprinting which relies on a time-frequency analysis of the actual signal and not the received measurement.

As you can see, ultrasonic sensor security has a wide range of applications which all aim to increase safety, protect against human life, and reduce the financial loss of being victim to an attack.

## 3. System Model

The system model assumes an ultrasonic sensor system on chip devices commonly used in automotive applications [9]. The sensor does the signal conditioning and processing for the transducer echo signals and transmits the distance to the obstacle and other parameters over the chosen interface like CAN, LIN. The on-board ECU allows complete configurability for the end applications.

The proposed fingerprint extraction happens on the sensor itself, during an initial calibration phase where the sensor learns the fingerprint and trains a model to identify its own echo and differentiate it from others. This model can be used at a later stage to identify if the sensor is under attack. Shown in Fig 1 is the block-diagram of the system model. During the calibration phase, the system learns its echo and trains the model which is then used to determine authenticity of the received signal. Specifically, the received signal is analyzed for fingerprint extraction in the background while the data gets processed to detect obstacles. The output from the sensor includes a validity flag along with the data to assure that the data is authentic and not subject to physical attacks. In the proposed framework, we use a power spectrum coefficients as features and a simple Gaussian Naive Bayes classifier to perform supervised learning and classification of labeled data. As the Naive Bayes classifier supports multi-class classification, it will not only allow our system to accurately detect when an attack occurs but also on what sensor, since most vehicles which utilize ultrasonic sensors use more than one.

Our system for combating attacks launched by the adversary is under the assumption that the time in which we detect an attack is not a leading factor in the success of our model. In real-time applications, ADAS systems have stringent safety requirements such as brake engagement that have a maximum latency of 0.1 seconds [10].

### 3.1. Data Model

Our data model assumes that the data inputs have the following characteristics as noted in [7] except we define the transmitted and received signals with the inclusion of noise characteristics emitted by the transducer due to a hardware fingerprint. We can describe the transmitted waveform of our ultrasonic sensor as an ideal sinusoidal signal

$$s(t) = A\cos(\omega_c t), \quad t\epsilon[0,\infty] \tag{1}$$

Where in Eq(1), $A$ is the amplitude of the signal, $t$ is the time and $\omega_c$ is the radial frequency of the carrier signal.

In reality, the transmitted signal will have some noise component to it as a result of the hardware fingerprint

$$s_r(t) = A\cos(\omega_c t) + n_r(t), \quad t\epsilon[0,\infty] \tag{2}$$

Where in Eq(2), $n_r(t)$ denotes the noise of the transmitted signal due to the hardware fingerprint.

At the receiver, the transmitted signal appears as

$$r(t) = \alpha\cos((\omega_c t+\omega_D)(t-\tau)+\theta)+n_r(t)+n(t), \quad t\epsilon[0,\infty] \tag{3}$$

Where in Eq(3), $\alpha$ represents the attenuated amplitude of the transmitted signal, $\omega_D$ is the Doppler velocity, $\tau$ is the time delay (time for the echoed signal to reach the receiver), $\theta$ is the phase shift, and $n(t)$ is the additive noise component. We expect $n_r(t)$ to be centered at the resonance frequency of our sensors since ultrasonic sensors transmit pulses by exciting a piezoelectric transducer [11]. This transducer will vibrate acoustically at the same frequency as the AC voltage

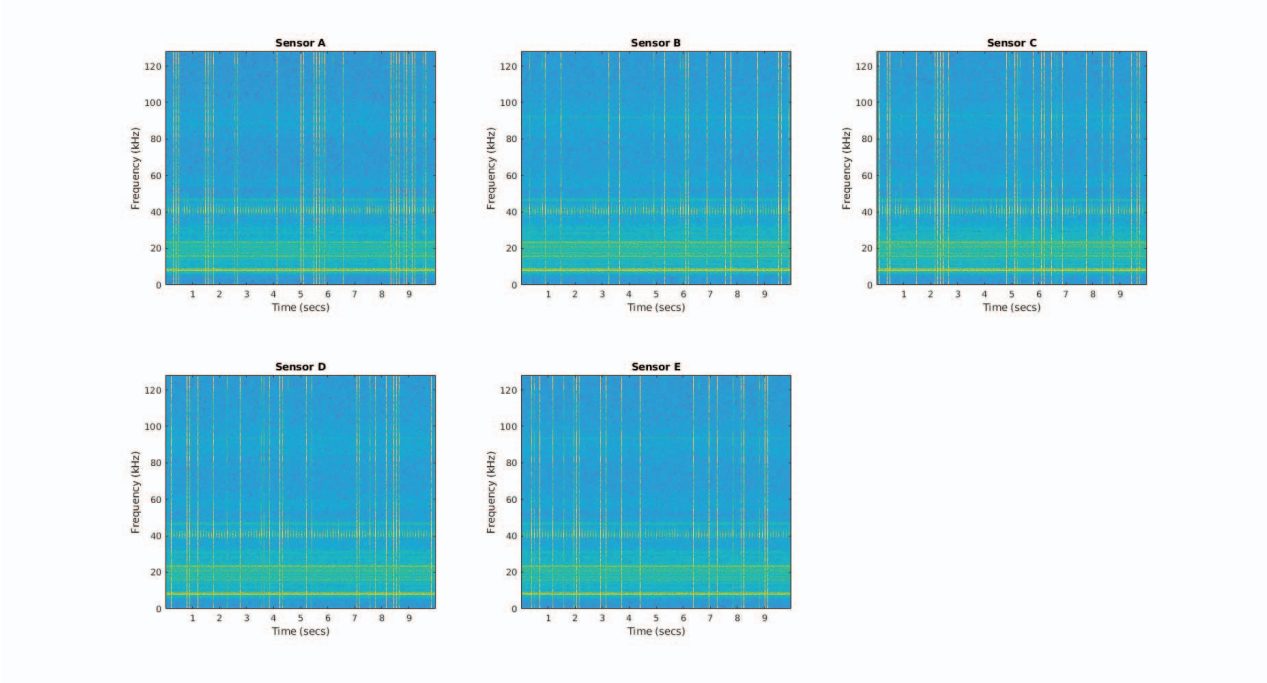


Figure 2: Spectrograms of sensor transmissions generated using 8 ms window size, 25% overlap, and Hanning weight window

that is applied to it. Noise due to microscopic hardware imperfections will be exacerbated around the resonance frequency of the sensor. Signals of this type will be analyzed and used as input to our classifier.

### 3.2. Threat Model

While evaluating the security of our model, it is important first to identify what possible adversaries we must defend against and what types of attacks they can employ. We identified these main physical channel attacks on the ultrasonic sensors. It is assumed that the attacker will be able to perform these three types of attacks and for launching these attacks, the assumption is made that the attacker will have a know-how of all the information of our system, such as what sensors are used, the frequency at which data is recorded, and even our method for defending against malicious attacks.

i) *Jamming Attacks*: The attacker will be able to perform jamming attacks [12], where the transducer of an ultrasonic sensor is always excited with ultrasound in such a way that it cannot measure the echo of its own transmitted ultrasonic waves and therefore cannot accurately perceive its surroundings.

ii) *Spoofing Attacks*: The attacker will be able to generate ultrasonic pulses to excite the transducer of an ultrasonic sensor such that a ”phantom object” can be perceived by the sensor when it is not truly there. This is the case when an ultrasonic wave is spoofed to the transducer of an ultrasonic sensor before the echo of its own transmitted wave can return, resulting in the sensor perceiving a non-existent object. Although this is difficult to perform while a sensor is in motion due to timing dependencies, it has been implemented on stationary sensors used in automobiles in [6] in the case where the attacker has knowledge of the frequency of ultrasonic sensor readings, which fits this threat model.

iii) *Sensor Damage & Replacement*: In addition to jamming and spoofing attacks, the adversary may also perform an attack that requires physical contact with the sensor. This is the case when the adversary damages [13] the sensor or replaces it entirely. It is assumed that the adversary is able to do this stealthily, such that visually it is not possible to tell whether or not a sensor has been physically damaged, replaced, or altered in any way.

The proposed framework can handle jamming and spoofing attacks along with the sensor damage contact-based attacks. Since we assume a smart sensor that runs the data-processing on-board, we cannot detect the sensor replacement contact-based attack.

## 4. Fingerprint Extraction

To extract and localize the hardware-specific fingerprints, we chose time-frequency analysis method. As spectrograms give the time-frequency distribution of time series data, we started with spectrogram analysis of the sensors under test. In Fig 2, the spectrograms of the five sensors

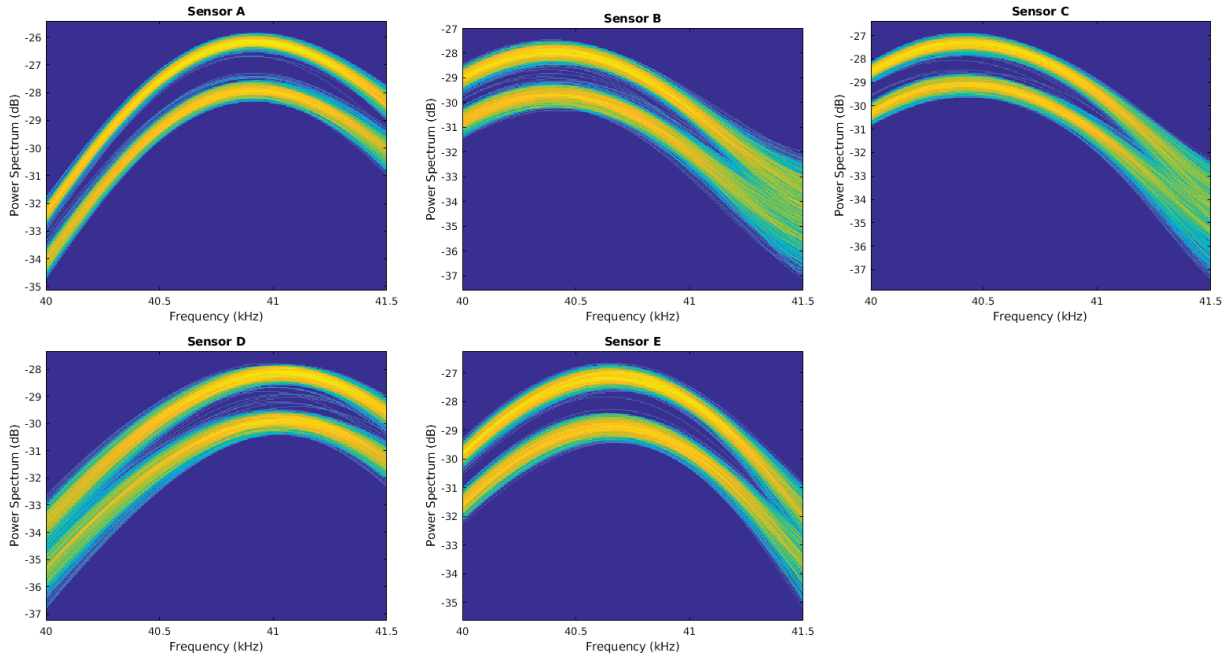


Figure 3: Power spectrum of sensors under test at 25 cm distance measurement

under test are shown. With the reduced window size of 8 ms, the frequency distribution of each sensor is visually distinguishable, although minutely, and laid the first step towards our claim towards the presence of an intrinsic sensor fingerprint. As a next step we focused on the spectral components at central frequency of operation of the sensors. We obtained the power spectrum of the ultrasonic sensor signal around the operating frequency of 40 KHz with a timing resolution of 250 ms and a frequency resolution of 1 KHz. The power spectrum is generated with persistence option to visualize the percentage of time that a particular frequency component is present in the input signal. The results as shown in Fig. 3, display a distinct feature in the form of the power spectrum peak location that can be used to identify each sensor. The power spectrum peak and the corresponding peak shape profile occurred at different frequencies for different sensors under test. It can be observed from the Fig. 3, that the spectral peaks for sensors under test, A,B,C,D & E occurred at 40.91, 40.36, 40.45, 41.03, 40.65 KHz respectively. The peak locations of any two sensors were separated with a 100 Hz frequency resolution and the peak roll off rates for different sensors are different as-well. Given the fact that our sensors under test are from same manufacturer, of same grade and data collection conditions are same across multiple experiment runs, the variation in the location of peak for power spectral components can definitely be considered as a unique fingerprint for each sensor. We used this variation in the peak location and the shape profile information as our main feature for the classification of the sensors. Though it can be argued that as the number of sensors increases drastically the frequency resolution might not be sufficient to distinguish different sensors based on the just the spectral peak location, for our end application of supporting Advanced Driver Assistance System (ADAS) or Automated Driving (AD) features, the number of ultrasonic sensors used in a vehicle is usually less than 15. For instance, Tesla autopilot advanced sensor coverage has the 12 ultrasonic sensors [14]. We observed similar trends in power spectrum peak location and shape profile at different distances as shown in Fig. 4. The power spectrum visualization in Fig. 4 shown in a table form with each row displaying the power spectra of a single sensor collected at different distances and similarly the columns represent the power spectra of different sensors at a given distance. While the peak location was a good feature to classify different sensors at a given distance it did not generate good results for distance agnostic sensor classification. It can be observed that for distance agnostic sensor classification feature the peak roll off rate and the shape profiles need to be used and modeled. This is considered as future extension of this research.

### 4.1. Dataset

In order to use fingerprinting as a means to classify which ultrasonic sensor a given waveform originated from, data from multiple sensors must be collected. As shown in the Fig. 5, an anechoic chamber was built out of acoustic foam in order to reduce reflections off of the surrounding environment from being recorded by the high-frequency microphone. The acoustic foam used has a high absorption

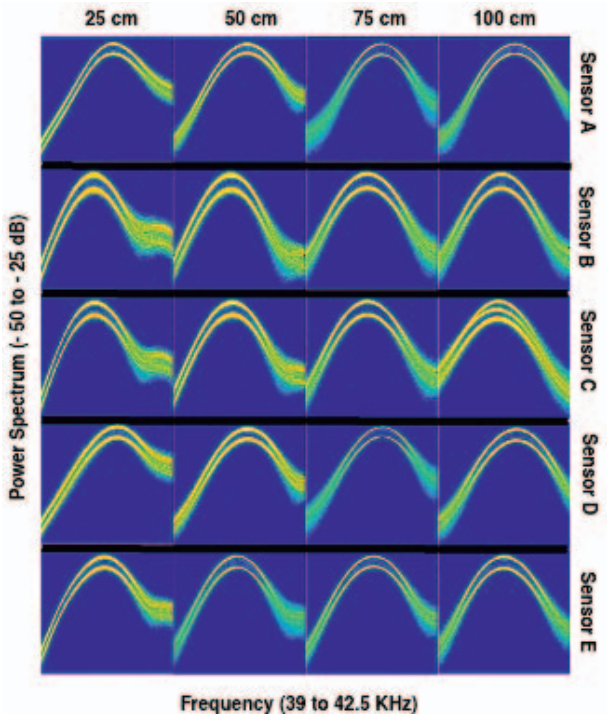


Figure 4: Power spectrum of all sensors under test at different distances

coefficient, which allows it to reduce the number of reflected sound waves that contact it. By reducing reflections of ultrasonic signals at the microphone, we can isolate singular pulses for comparison and analysis for our system. The ultrasonic sensor, microphone, and power circuit needed to operate the ultrasonic sensor were placed across from each other inside of the anechoic chamber on top of 3D-printed mounts, which were also covered in acoustic foam. Once the recordings began, they were left to run for ten minutes. The first thirty seconds and last thirty seconds of each recording were removed to ensure no audio from external sources was recorded by the microphone because of operators leaving and entering the room where the chamber was located. Five MB1013 HLRV Max Sonar sensors were analyzed and used for fingerprint classification. Data was captured at specific distances of 25cm, 50cm, 75cm, and 100cm.

### 4.2. Feature Vector Generation

Feature vector generation was done automatically by our own program via MATLAB. Our algorithm takes in time-series sensory data and analyses its time varying frequency content as in Fig 2. Since we are using a Gaussian Naive Bayes classifier, we anticipate that our feature vectors will have a normal distribution. We will consider a feature vector being the energy of a given frequency bin over the sampling time. For example, a feature vector could be the energy in the 40KHz frequency "bin" over time, since it will have a normal distribution and that distribution will later be proved to be different between sensors. However, the distribution in a low energy frequency interval will have very little variation and be similar between sensors. At some point, the amount of feature vectors, or frequency bins considered will have diminishing returns if too large. The amount of feature vectors which maximizes the accuracy of our classifier is an optimization problem. Using the optimization toolbox in MATLAB, where the goal was to maximize our testing accuracy under the constraint that the accuracy must be at least 90%, we found that the amount of feature vectors does not vary with distance and is approximately 161 vectors. This corresponds to $\pm$ 1kHz above and below the determined resonance frequency.

With this frequency profile, the vectors used as input to our algorithm are related to the frequency distribution displayed in Fig 3. To ensure that the distribution of energy in each frequency bin was discernible between sensors as we hypothesized, we plotted these distributions for each frequency bin. Fig 6 proves such distributions exhibit the desirable features.

**Algorithm 1** Feature Extraction algorithm (MATLAB)

**procedure** FEATURE($S$) ▷ .wav files for n sensors
  $F_s = 256kHz$
  **for** $i = 1 : n$ **do**
    $A_i = audioread(S_i)$
    $M_i = abs(spectrogram(A_i))$
    $Nrows = size(M_i, 1)$
    $Peak = max(M_i(1 : Nrows), [], 2)$ ▷ Peak of each frequency bin
    $MaxPeak = max(Peak)$
    $index_i = find(M_i == MaxPeak)$
  $PeakBin = round(avg(index))$
  $UBound = PeakBin + a$
  $LBound = PeakBin - a$
  **for** i=1:n **do**
    $\tilde{M}_i = M_i(LB : UP, :)$
  $Result = [\tilde{M}_1\tilde{M}_2...\tilde{M}_n]$ ▷ Horizontal Concatenation
  **return** $Result$

### 4.3. Classification

The Gaussian Naive Bayes classifier works by using Bayes Theorem [15] which explains the conditional probability of an event occurring based on the existence of features or other events being present. Mathematically, this is stated as

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x-1, ..., x_n)} \quad (4)$$

where y is the class variable and the feature vector is described as x.

The Naive Bayes classifier assumes conditional independence of features, hence we can say

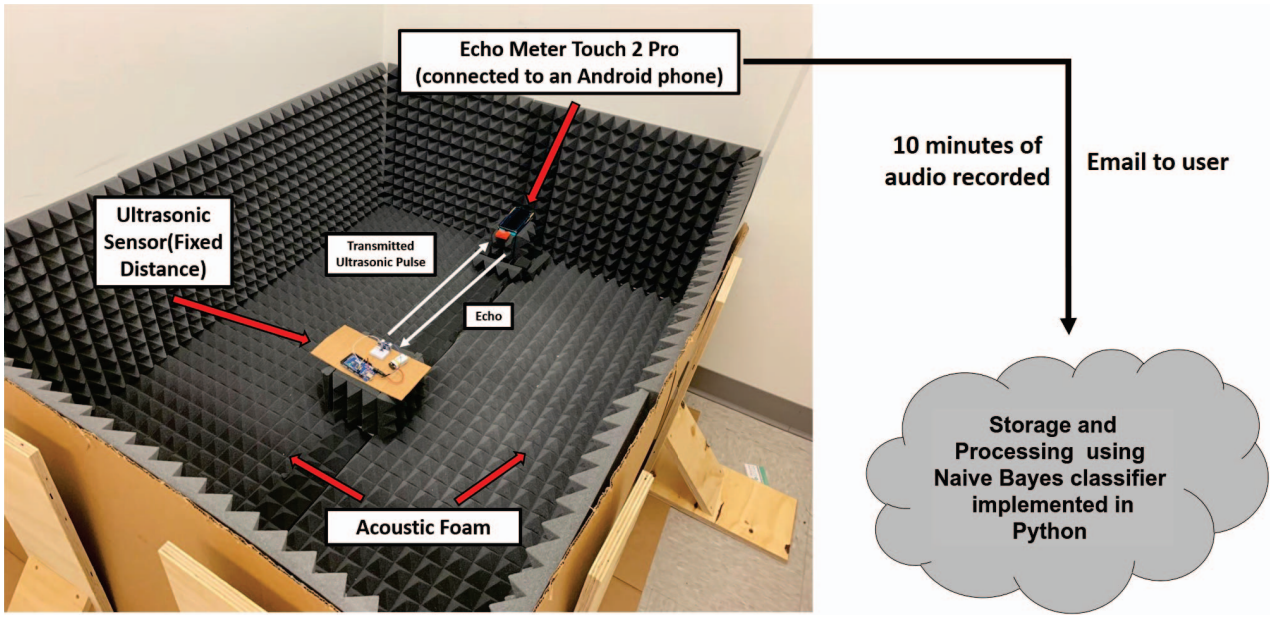


Figure 5: Data collection set-up for fingerprint extraction

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \tag{5}$$

In regards to all feature vectors, this reduces to

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{6}$$

The denominator of this equation can be assumed constant given the input data and the classification becomes

$$P(y|x_1, ..., x_n) \ \alpha \ P(y)\prod_{i=1}^{n} P(x_i|y) \tag{7}$$

Where alpha denotes proportionality. Using the Maximum A Posteriori [16] estimation, we can identify which class maximizes the classification rule, and our decision becomes

$$\hat{y} = arg \ max_y P(y)\prod_{i=1}^{n} P(x_i|y) \tag{8}$$

In a Gaussian Naive Bayes classifier, the probability of a feature vector being present given a class variable is expressed as

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{9}$$

Where $\sigma$ and $\mu$ are maximum likelihood estimation parameters.

In our implementation, the class variable y represents the five different sensors whose fingerprints were extracted. The feature vectors for each of the sensors is the amount of energy in a frequency bin over the duration of our ten-minute audio samples. Our hypothesis is that the Naive Bayes classifier will prove effective in differentiating between sensors because the existence of a hardware fingerprint will emerge in minuscule variations in transmitted signal strength in various bins. Our reasoning is supported by histograms for the energy distributions in different frequency bins for the five sensors. From the spectrograms in Fig. **??**, we can see that most of the energy content is around 40 kHz. When analyzing the energy distributions of the five sensors in the 40 kHz frequency bin 6, we can see the dissimilarity between distributions motivates the idea that a Gaussian Naive Bayes classifier will prove effective when several frequency bins are added as feature vectors for our classifier.

A MATLAB script was written to find the energy bin with the maximum energy content. From there, the frequency bins corresponding to 2kHz above and below the maximum energy bin were used as feature vectors for our classifier. In total, 161 feature vectors were used. Using the bins with the highest energy content filtered out low energy bins where the distribution (mean and standard deviation) of energy was very similar between sensors.

## 5. Experiments & Results

The first step in building a system model to counter the physical attacks on an ultrasonic sensor is to establish that different ultrasonic sensors generate fingerprints in their transmissions unique to the host and this finger print can be used to identify the host sensor. To prove this point, we set-up an experiment as shown in Fig. 5. The microphone placed

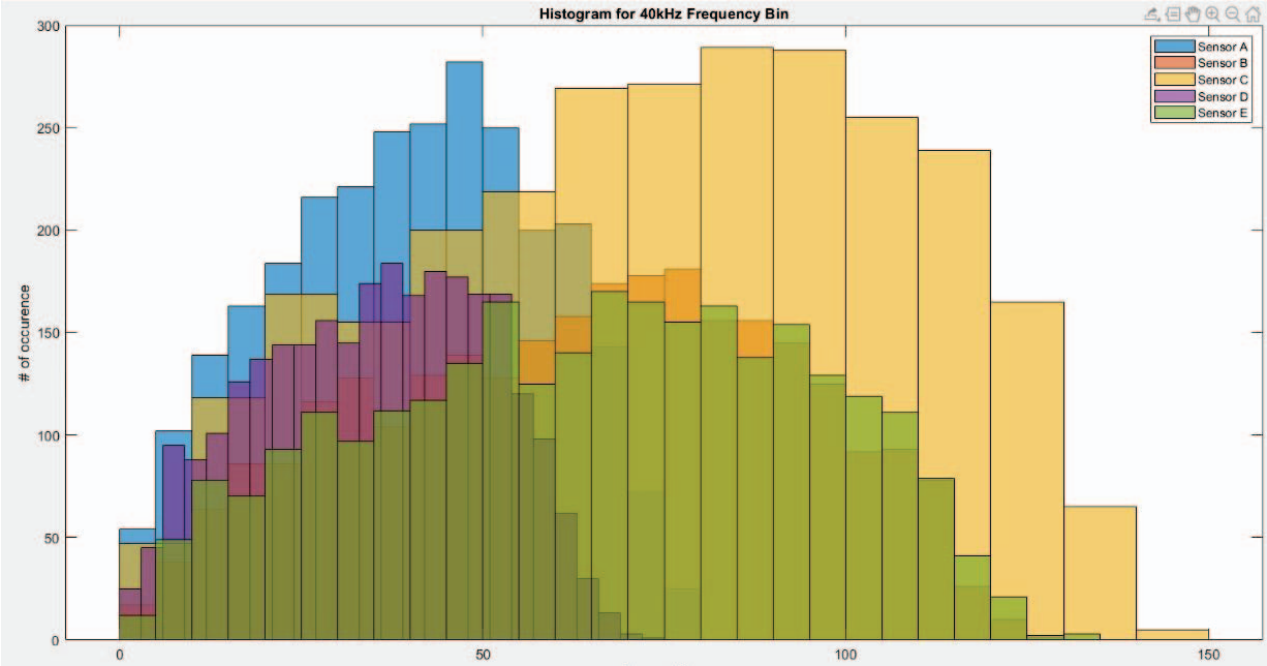


Figure 6: Energy Distributions with differing means and standard deviations

at various distances from an ultrasonic sensor captures the sensor transmissions and records them.

This recorded data is then used to generate feature vectors from spectrograms. After generating the scattering features for each ultrasonic sensor under test, the Gaussian Naive Bayes model is trained with the training dataset.

The Gaussian Naive Bayes classifier also had promising results. Data used from the same experiment shown in Fig. 5. was input to our classifier. One benefit of the Gaussian NB method is that only 10 percent of the data was needed for training to achieve high accuracy classification.

Table 1: ACCURACY - GAUSSIAN NB

| Distance (cm) | Training Size | Test Size | Accuracy |
|---|---|---|---|
| 25 | 10% | 90% | 99.67% |
| 50 | 10% | 90% | 96.68% |
| 75 | 10% | 90% | 95.42% |
| 100 | 10% | 90% | 99.66% |
| Mixed Distances | 10% | 90% | 91.72% |

As an extension, we decided to synthetically saturate the received signal of our ultrasonic sensor by adding a percentage of the peak noise values seen graphically in Fig 2 as a DC component to the signal before the spectrogram is applied. By synthetically adding gaussian white noise [17], the discernibly of the fingerprint was diminished. The goal of this was to experiment with pseudo-jamming to see at what point our classifier would no longer be able to successfully identify a sensor.

To recursively add noise until the fingerprint was no longer identifiable, we let the amount of saturation be proportional to some value of the peak value.

$$N_r[n] = x[n] + \alpha\sigma_x N[n] \tag{10}$$

Where $x(n)$ is the received digital signal, $\sigma_x$ is the standard deviation of the original signal, $\alpha$ is a saturation coefficient and $N[n]$ is a noise signal with standard normal mean and standard deviation and $N_r[n]$ is our total saturation which is added to the entire time-domain signal. Fig 7 shows the affect different values of $\alpha$ have on the spectrum of the received signal.

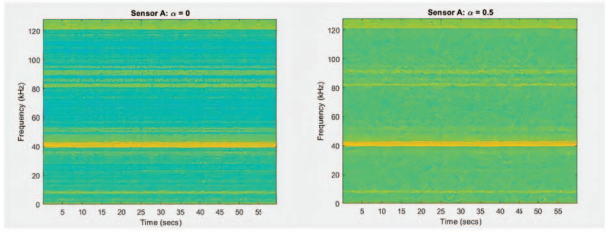


Figure 7: Saturation of Received Signal - Spectrogram Visual

The classifier performed reasonably well for values of $0 \leq \alpha \leq 0.539$ The table below shows the accuracy of the classifier for different values of $\alpha$ for a single sensor

Table 2: ACCURACY - SATURATED GAUSSIAN NB

| Distance (cm) | $\alpha_{max}$ | Accuracy |
|---|---|---|
| 25 | 0.539 | 91.41% |
| 50 | 0.4179 | 93.38% |
| 75 | 0.4009 | 92.59% |
| 100 | 0.394 | 90.33% |
| Mixed Distances | 0.2154 | 91.72% |

## 6. Conclusion and Future Work

In this paper, we presented the concept of using the host sensor's inherent characteristics or distortions called fingerprints to identify itself. Using the ultrasonic sensor set, we demonstrated that the proposed fingerprint extractor and the classifier framework could identify the host sensor successfully with a minimum of 96% accuracy in the absence of synthetic noise. Therefore, we establish that the sensor intrinsic distortions can be successfully used to identify them. The framework is being evaluated for other complex scenarios like receiving target echoes from single and multiple sensors. Further tests to analyze the performance of the countermeasure framework on data collected under different scenarios is being evaluated along with other lightweight fingerprint extraction techniques. We intend to extend collect more data and to support and classify ten different sensors. Future work for this project also aims to observe the results of the proposed method on actual jamming attacks as well as spoofing attacks. For real time applications, it is desired that the system's timing performance is monitored to see if it meets the requirements for in vehicle deployment. It is also pertinent that an ensemble classifier is trained and implemented to detect ultrasonic fingerprints at intermediate distances, such as between 25cm and 50cm in our current framework.

## References


[1] H. Shin, Y. Son, Y. Park, Y. Kwon, and Y. Kim, "Sampling Race: Bypassing Timing-Based Analog Active Sensor Spoofing Detection On Analog-Digital Systems." *Woot*, 2016. [Online]. Available: https://www.usenix.org/system/files/conference/woot16/woot16-paper-shin.pdf

[2] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *IN: Proceedings of the workshop on embedded security in cars (ESCAR)'04*, 2004.

[3] T. Bapin and P. Mov'eo, "advanced driver assistance system – nexyad." [Online]. Available: https://nexyad.net/Automotive-Transportation/?tag=advanced-driver-assistance-system

[4] S. Deng, Z. Huang, X. Wang, and G. Huang, "Radio frequency fingerprint extraction based on multidimension permutation entropy," *International Journal of Antennas and Propagation*, vol. 2017, p. 1–6, 2017.

[5] C. M. Ahmed, A. Mathur, and M. Ochoa, "Noisense: Detecting data integrity attacks on sensor measurements using hardware based fingerprints," 2017.

[6] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, no. 8, p. 109, 2016.

[7] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5015–5029, 2018.

[8] C. M. Ahmed and A. P. Mathur, "Hardware identification via sensor fingerprinting in a cyber physical system," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2017, pp. 517–524.

[9] T. Instruments, *TI Designs: TIDA-00151 Automotive Ultrasonic Sensor Interface for Park Assist or Blind Spot Detection Systems*. Texas Instruments, Texas Instruments Incorporated, 2014.

[10] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving," *ACM SIGPLAN Notices*, vol. 53, no. 2, p. 751–766, 2018.

[11] N. W. Hagood and A. von Flotow, "Damping of structural vibrations with piezoelectric materials and passive electrical networks," *Journal of Sound and Vibration*, vol. 146, 1991.

[12] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 1307–1315.

[13] N. Elvin *et al.*, ""a self-powered damage detection sensor," " *The Journal of Strain Analysis for Engineering Design*, vol. 38, no. 2, Feb. 2003.

[14] *MODEL S OWNER'S MANUAL*. Tesla, 2020.

[15] H. ZHANG, "Exploring conditions for the optimality of naÏve bayes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 02, pp. 183–198, 2005. [Online]. Available: https://doi.org/10.1142/S0218001405003983

[16] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[17] M. K. Simon, *Probability distributions involving Gaussian random variables a handbook for engineers and scientists*. Boston, Springer US, 2002.