electronics

MDPI

Article

Design of a Cyberattack Resilient 77 GHz Automotive Radar Sensor

Onur Toker * and Suleiman Alsweiss

Department of Electrical and Computer Engineering, Florida Polytechnic University, Lakeland, FL 33805, USA; salsweiss@floridapoly.edu

* Correspondence: otoker@floridapoly.edu

Received: 9 March 2020; Accepted: 22 March 2020; Published: 28 March 2020

Abstract: In this paper, we propose a novel 77 GHz automotive radar sensor, and demonstrate its cyberattack resilience using real measurements. The proposed system is built upon a standard Frequency Modulated Continuous Wave (FMCW) radar RF-front end, and the novelty is in the DSP algorithm used at the firmware level. All attack scenarios are based on real radar signals generated by Texas Instruments AWR series 77 GHz radars, and all measurements are done using the same radar family. For sensor networks, including interconnected autonomous vehicles sharing radar measurements, cyberattacks at the network/communication layer is a known critical problem, and has been addressed by several different researchers. What is addressed in this paper is cyberattacks at the physical layer, that is, adversarial agents generating 77 GHz electromagnetic waves which may cause a false target detection, false distance/velocity estimation, or not detecting an existing target. The main algorithm proposed in this paper is not a predictive filtering based cyberattack detection scheme where an "unusual" difference between measured and predicted values triggers an alarm. The core idea is based on a kind of physical challenge-response authentication, and its integration into the radar DSP firmware.

Keywords: automotive radar sensors; cybersecurity; attack resilient radar DSP

1. Introduction

During the last decade, various levels of connectivity and autonomy gained significant importance in automotive industry, and this trend triggered large research projects both in the academia and in the industry [1,2]. There are a multitude of key developments behind this trend which include advancements in electronic, communication, and remote sensing technologies to increase efficiency and improve safety and reliability. However, these key technologies also introduced new and quite challenging problems, and cybersecurity for autonomous vehicles (AV) is among the most important ones. In this paper, we will focus on physical layer cyberattacks to AV radar sensors. In an AV system, radars are typically used for target detection, and range/velocity estimation, and AV algorithms use this information for steering decisions [3,4]. Any kind of cyberattack to an AV system may cause serious and/or fatal accidents.

Cyberattacks can be of passive type where an adversarial agent simply listens to the information received by the sensor, or of active type where an adversarial agent generates physical signals to spoof the sensor [5]. All AV radar sensors, whether isolated or part of a sensor network, are susceptible physical layer attacks. For example, we may have an autonomous vehicle equipped with a 77 GHz radar, and a secondary 77 GHz transmitter may generate electromagnetic waves which may cause a false target detection, false distance/velocity estimation, or not detecting an existing target. If the autonomous vehicle is also a part of a connected AV system, we can have cyberattacks at the network/communication layer too. For a network/communication layer type attack, the adversarial agent intercepts data packets, and/or injects modified data packets to cause harmful effects. In this paper, we focus on active cyberattacks at the physical layer. As a side note, we would like to cite Reference [6] where a closely related problem of interference for AV radars is investigated.

Because of the popularity of Frequency Modulated Continuous Wave (FMCW) radars in the automotive industry, we will assume a standard FMCW radar radio frequency (RF) front-end, and propose DSP algorithms at the radar firmware level for cyberattack resilience. Improved attack resilience can also be achieved by using innovative RF front-ends, but these nonstandard radar systems will be more costly and hence may not be adopted by the automotive industry. In summary, the main question that we are trying to answer is how to improve the cyberattack resilience by using a standard FMCW radar RF front-end by using innovative radar DSP implemented at the firmware level.

There are various statistical techniques that can be used for cyberattack detection. Basically, a measurement which "looks" like an out-liar is a warning sign, and this basic principle can be used to develop quite effective algorithms, for example see References [7–10] and references therein. There are also numerous papers where a prediction filter is used for some kind of estimation, and an alarm is triggered if an "unusual" difference is detected between measured and predicted values, for example References [11–13] and references therein. This prediction filter can be a Kalman filter, an artificial intelligence (AI) based estimator, or something completely different. A detector measures the difference between the estimate and measurement, and compares this difference with a threshold value. If the difference is above a set threshold, a cyberattack alarm is generated. If an attack is detected, the system may simply ignore some of the latest measurements, and may continue to operate using only the estimated values. This general approach is demonstrated to be effective in a good number of cases. However, slow-attacks, that is, for cyberattacks where an intentional error is introduced at a relatively small rate so that the difference measured by the detector is always below the threshold [14], is a known weak point of these techniques. In general, any system is subject to certain measurement noise, and to minimize false alarms, the threshold must be set to a value bigger than a certain minimum. However, if the injected error is small, and difficult to distinguish from "natural" system noise, it will be hard to detect but may easily cause a gradually increasing unstable behavior.

The technique proposed in this paper is not a prediction filter based approach, indeed it is inspired from the physical challange-respone authentication (PyCRA) method discussed in Reference [15]. As a side note, we would like to emphasize that using the proposed physical layer cyberattack detection system implemented at the firmware level, together with a prediction based cyberattack detector implemented at higher layers may result improved cyberattack resilience. As a related work, we also would like to cite References [16,17] where a modulation based PyCRA system is proposed. Although the techniques developed in References [15–17] are applicable to many different sensors, they are all based on turning of the sensor's physical output signal off at random instants/periods. For better cyberattack detection, sensor's physical output must be turned on for shorter and randomly selected periods of time. But when the sensor is off, we are unable to collect data, which degrades the sensor's accuracy and/or responsiveness. Of course, this effect can be partially mitigated by using certain prediction techniques. What is proposed in this paper is specific to FMCW radars, and is not based on turning off the radar sensor's transmitter during randomly selected intervals. The key idea is randomly changing the sign of the sweep. An attack signal with a different sweep sign will result a false target at a negative distance, and hence can easily be distinguished.

In this paper, we will propose two integrated algorithms, the first one is for cyberattack detection, and the second one is for attack resilient radar DSP. The radar DSP basically generates 2D heat-maps for range/velocity estimation. Our design objectives are

(A) False cyberattack alarm rate, P_F, must be low
(B) Cyberattack miss rate, P_M, must be low
(C) The radar DSP output must be robust against undetected cyberattacks

Among these three objectives, the first two are obvious, and the last one basically means that if a cyberattack is below the detection threshold, its effect on radar DSP should be small. This work is related to the authors' previous simulation based study [18] where all of the above objectives are taken into account for a different algorithm. In this work, we have a significantly improved algorithm which is tested using real data obtained by utilizing the techniques developed in References [19–21].

Our results are consistent with the simulation based approach of Reference [18]. Basically, false alarm rate of the system, P_F, seems to be extremely small to measure experimentally, therefore we indirectly estimate this small probability using certain other measured parameters. The probability of miss, P_M, depends on the cyberattack signal power. For very "weak" cyberattacks, probability of miss P_M can be quite high, but their effect on radar DSP performance will be also very small, see Section 7. We carefully study cyberattacks which are just below the detection threshold, and hence not detected, that is, missed. Our attack resilient radar DSP algorithm is designed to be robust for such cases, and it will be shown that the effect of such cyberattacks which are just below the detection threshold are comparable to the effect system noise, see Section 7. This is probably the most important observation made in this work.

The remainder of this paper is organized as follows: In Section 2, we will describe general FMCW radar basics and Texas Instruments (TI) AWR series radars used in this work. In Section 3, high level description of the proposed system is presented, cyberattack detection and attack resilient radar DSP algorithms are given in pseudo code format. In Section 4, we present our experimental setup, and how data is collected for different attack scenarios. In Section 5, we analyze system noise, which affects both the radar performance, and the cyberattack detection threshold. In Section 6, we analyze the cyberattack detection algorithm under various attack scenarios. Finally, in Section 7, we study the performance of the proposed radar DSP algorithm for cyberattacks which are just below the detection threshold.

2. Review of FMCW AV Radars

An FMCW AV radar may have multiple transmit and receive antennas, and this multi-input/multi-output (MIMO) architecture can be used for beam forming and direction finding. One of the most critical components of an FMCW radar is the voltage controlled oscillator (VCO) with output frequency changing with input voltage. Most VCOs exhibit some nonlinear behavior, but various non-linearity correction methods are known to mitigate this problem, see References [22,23] and references therein for details.

The block diagram of a standard FMCW radar with a single transmit (TX) and single receive antenna (RX) is given in Figure 1.

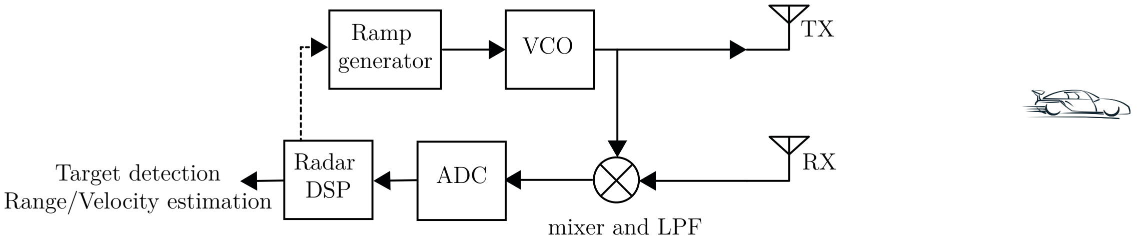


Figure 1. Architecture of an Frequency Modulated Continuous Wave (FMCW) radar.

The block labeled as mixer and low-pass filter (LPF) multiplies the transmitted and received signals, and outputs the low-pass filtered result to the analog to digital converter (ADC) block. After that, we have the radar DSP block which is responsible for target detection, and range/velocity estimation.

The transmitted signal $S_T(t)$ can be written as

$$S_T(t) = A_T(t) \cos \left(2\pi \int_0^t f_T(v_i(\tau)) d\tau + \theta_T \right), \tag{1}$$

where θ_T is a constant, $A_T(t)$ is the VCO output amplitude, and f_T is the instantaneous VCO output frequency as a function of the VCO control input voltage v_i. We assume that one of the known VCO non-linearity correction methods is already implemented, and hence $f_T(v_i) = f_0 + B(v_i/v_{i,max})$, where f_0 is the initial VCO frequency, $v_{i,max}$ is the maximum VCO control input voltage, and B is the VCO bandwidth. If there is an object at distance d from the radar, the received signal, $S_R(t)$, will be a delayed and possibly attenuated copy of the transmitted signal. More precisely, $S_R(t) = S_T(t-h)$, where $h = 2d/c$, d is the target distance, and c is the speed of light. In this case, the output of the mixer and LPF will be the beat signal, $S(t)$, where

$$S(t) = A(t)\cos\left(2\pi f_T(v(t))\frac{2d(t)}{c} + \theta\right), \tag{2}$$

where θ is a constant, and $A(t)$ is the beat signal amplitude.

Depending on the sign of the VCO sweep, $s \in \{-1, +1\}$, that is, the sign of the ramp input to the VCO, beat signal frequency will be

$$f_b = \frac{2Bd}{t_d c} + s\frac{2f_0 v}{c}, \tag{3}$$

where d is the target distance, v is the target velocity, t_d is the sweep duration, and B is the sweep bandwidth. If the VCO is driven by a triangular signal having both positive and negative ramps, spectral analysis of the beat signal for positive and negative ramps can be used to estimate the target distance and velocity simultaneously.

Although this 1D spectral approach works quite well if there is a single reflector, alternative techniques are necessary for crowded environments with multiple reflectors. More specifically, for an AV related application, if we would like to detect moving objects with relatively smaller radar cross section, a different 2D FFT based technique can be used, see References [24–28] and references therein. However, there are certain performance limitations of FMCW radars [19,24,25]. If the beat signal is sampled at frequency f_s, the resolution in target distance estimation (δd) will be

$$\delta d = \frac{c}{2B}, \tag{4}$$

which indicates that VCOs with higher bandwidth will have better distance (range) resolution. The maximum unambiguous range of an FMCW radar (d_{max}) will be

$$d_{max} = \frac{f_s c}{2S}, \tag{5}$$

namely, increasing the sampling frequency improves the maximum distance. For moving targets, the maximum unambiguous velocity will be

$$v_{max} = \frac{\lambda}{4t_d}, \tag{6}$$

where t_d is the chirp duration [19,24,25]. In an FMCW radar system, a frame is defined as M chirps combined together. Using 2D FFT techniques, velocity can be estimated with resolution

$$\delta v = \frac{\lambda}{2Mt_d}, \tag{7}$$

see References [19,24,25] and references therein. In Reference [29], the authors have a real-time demo video for 2D FFT based heat-map generation using TI AWR series radars.

Review of the Texas Instruments 77GHz Automotive Radar

In this section, we summarize some of the configuration parameters of the TI AWR1642 automotive radar used in this work, see Figure 2. All of our cyberattack detection tests, and all radar DSP performance tests are done using real data, and all experimental data is obtained by using this specific 77 GHz radar. Indeed to generate a realistic cyberattack, we used two of these radars units, see Figures 5 and 6. Although TI has a MATLAB based offline recording solution, we have used the real-time framework developed in Reference [19].

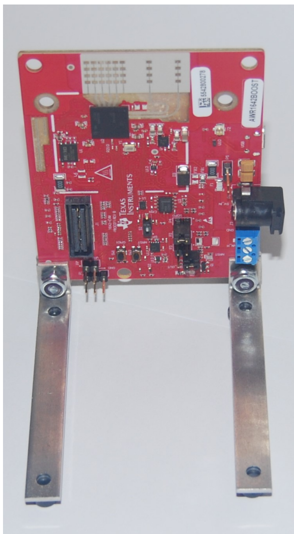


Figure 2. 77 GHz Texas Instruments automotive radar (AWR1642).

TI AWR1642 is an FMCW radar at 77GHz with a maximum bandwidth of 4 GHz. When it is used with the DCA1000 FPGA board, a single chirp will have $t_d = 160$ µs duration, which is repeated $N_r = 128$ times over a $T_m = 40$ ms time frame. Furthermore, during a single chirp, data is acquired at 10 MHz for 25.6 µs. See References [19,20] for our detailed real-time radar framework, and a sample AV application.

3. Cyberattack Detection System

In this section, we will describe the architecture of the proposed cyberattack detection system. Performance analysis using real data will be presented in the following sections.

Our high-level system block diagram is shown in Figure 3. For simplicity, the power amplifier before the transmit antenna, and the low noise amplifier after the receive antenna are not shown. Similarly, the low pass filter after the mixer is not included in the block diagram. To further simplify the discussion, we adopt the complex signal notation, and use a single ADC block after the mixer. If we use the real signal notation, there will be a 90° phase shifter, two mixer circuits, two low pass filters, and two ADCs for in-phase and quadrature components.

Compared to a standard FMCW radar system, the main difference of the proposed system is the random signature generator software block, which generates a random sequence of 1's and 0's. The ramp generator outputs a positive slope ramp for 1's, and a negative slope ramp for 0's. In other words, for each chirp, either a positive or a negative frequency slope is selected on a random basis. Note that, we are using a standard FMCW radar RF subsystem and hardware, and we have one extra software block for generating random signatures to control the ramp generator, and another extra software block for cyberattack detection DSP algorithm. In the proposed system, the AV sensor software block has a slightly modified DSP algorithm to suppress the effects of "weak" undetected attack signals, and detailed analysis of this will be presented in the following sections.

Consider an AV radar with N_{TX} transmit, and N_{RX} receive antennas. We assume that each transmit antenna is driven by the same VCO with optional phase shifters for beam forming, and each receive antenna has its own signal path consisting of its own mixers, and ADCs. Our focus will be on the $N_{RX} = 1$ case, because the proposed cyberattack Detector (Detection DSP) block shown in Figure 3 can be added to all (or some) of the receive signal paths as independent cyberattack detectors.

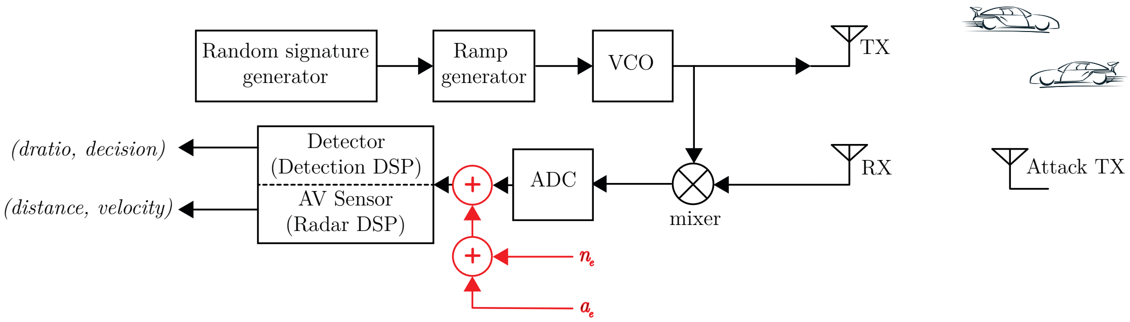


Figure 3. The proposed system block diagram. Signal path shown in red color represents mathematically equivalent effects of system noise and attack signals.

Mathematical Notation

In this section, we will summarize the mathematical notation used to describe the proposed system. As shown in Figure 4, the AV radar outputs a chirp signal in every t_d seconds. However, the slope of the chirp is based on the randomly generated signature, and can be either positive or negative. Positive slope chirps are shown in blue color, have increasing instantaneous frequency, and are marked with a 1 on the top. Negative slope chirps are shown in red color, have decreasing instantaneous frequency, and are marked with a 0 on the top. These non-overlapping chirp windows are denoted as $I_0, \cdots, I_{M-1}$, and we have a total of M chirps followed by a blank period of $T_m - Mt_d$ seconds. This time window of T_m seconds is called a measurement cycle, or frame duration, and the corresponding received data is called the frame data. Basically, the AV Sensor (Radar DSP) block shown in Figure 3, outputs an estimate about the environment in every T_m seconds. This estimate usually consists of a set of identified targets (or reflectors), their distance and velocity values. In most radar applications, this is done by using the 2D FFT of the frame data, and some additional post-processing.

We now introduce some relevant notation for random signatures. Let $\mathbb{Z}_M$ be the set $\{0, \cdots, M-1\}$. The set of all functions from $\mathbb{Z}_M$ to $\{0,1\}$ will be denoted by $\{0,1\}^{\mathbb{Z}_M}$. Basically, a random signature is a function in $\{0,1\}^{\mathbb{Z}_M}$ with some extra conditions. More precisely, a $\rho \in \{0,1\}^{\mathbb{Z}_M}$ is a random signature iff it is equal to 1 for $M/2$ values in $\mathbb{Z}_M$, and is equal to 0 for the remaining $M/2$ values in $\mathbb{Z}_M$. In other words, a $\rho \in \{0,1\}^{\mathbb{Z}_M}$ is as a random signature iff $\sum_{k \in \mathbb{Z}_M} \rho[k] = M/2$. The set of all possible random signatures will be

$$\mathcal{R} = \left\{ \rho \in \{0,1\}^{\mathbb{Z}_M} \; : \; \sum_{k \in \mathbb{Z}_M} \rho[k] = M/2 \right\}. \tag{8}$$

The reason behind this $M/2$ value is the inequality

$$\binom{M}{M/2} \geq \binom{M}{g}, \quad \text{if } g \in \mathbb{Z} \text{ and } g \neq M/2, \tag{9}$$

in other words, the $M/2$ which appears in the definition of $\mathcal{R}$ results in the largest set of random signatures, equivalently the largest size for the set $\mathcal{R}$.

Basically, for each frame a randomly selected element of $\mathcal{R}$ will be used as the random signature ρ. The slope of each chirp in a frame is determined by the value of ρ at the chirp index. In other words, if $\rho[k] = 1$, then the k^{th} chirp in the time window I_k will have positive slope, that is, increasing instantaneous frequency. However, if $\rho[k] = 0$, then the corresponding k^{th} chirp in the time window I_k will have negative slope, that is, decreasing instantaneous frequency. (See Figure 4).

The set $\mathcal{R}$ is indeed a very large set, because the cardinality of $\mathcal{R}$ is $\begin{pmatrix} M \\ M/2 \end{pmatrix}$, and

$$0.68\frac{2^M}{\sqrt{M}} \leq \begin{pmatrix} M \\ M/2 \end{pmatrix} \leq 0.86\frac{2^M}{\sqrt{M}}, \tag{10}$$

where we used the inequality $\sqrt{2\pi}\, n^{n+\frac{1}{2}}e^{-n} \leq n! \leq e\, n^{n+\frac{1}{2}}e^{-n}$. For $M = 128$, we have at least 2×10^{37} random signatures in $\mathcal{R}$. If the AV radar is used continuously for 100 years with 25 frames/s, we will use only

$$25 \times 3600 \times 24 \times 366 \times 100 \leq 8 \times 10^{10} \tag{11}$$

frames, which is less than 10^{-26} times the total number of random signatures in $\mathcal{R}$.

In our proposed design, the AV radar is operated according to the following procedure:

1. In every measurement cycle, a random signature $\rho \in \mathcal{R}$ is selected randomly.
2. Throughout the measurement cycle, there will be total M chirps with index $k = 0, \cdots, M-1$.
3. If $\rho[k] = 1$, the ramp generator outputs a positive slope ramp resulting in an up-chirp. Otherwise, the ramp generator outputs a negative slope ramp resulting in a down-chirp.

In Figure 4, a sample AV radar frame is shown. There are $N_r = 10$ chirp windows, and half of them, I_0, I_2, I_5, I_8, I_9, are up-chirps (shown in blue color), and the remaining ones are down-chirps (shown in red color). For each measurement cycle, we re-select a $\rho \in \mathcal{R}$. Note that, there are blank periods between successive chirps, which can be used for VCO relaxation. There are also blank periods between successive frames, that is, $T_m - Mt_d$, which can be used for on-chip signal processing of the frame data.

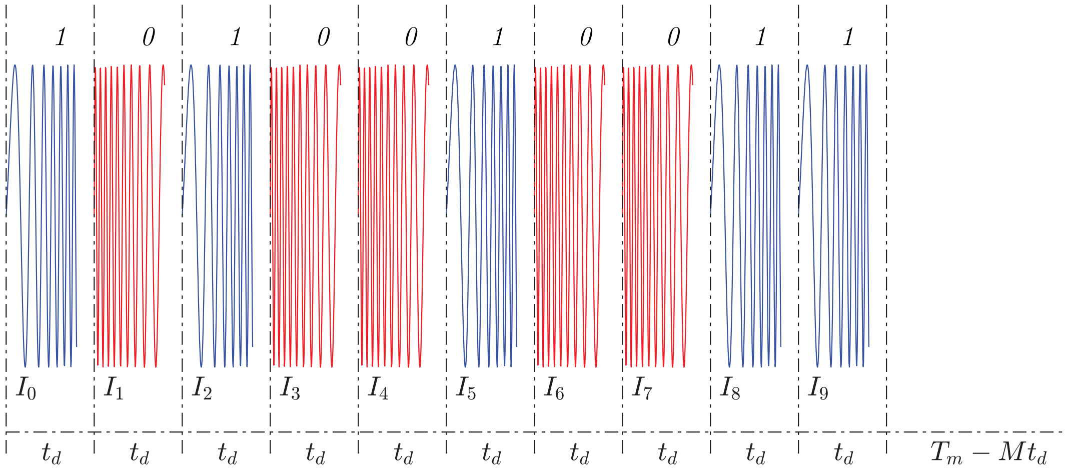


Figure 4. Measurement period of an autonomous vehicle (AV) radar for $M = 10$. Random signature values are shown on the top: A 1 represents an up-chirp (blue), whereas a 0 represents a down-chirp (red).

The signal observed at the input of ADCs, is denoted by $y_k[m]$, where $k = 0, \cdots, M-1$ is the chirp index, and $m = 0, \cdots, N_s - 1$ is the sample index. Our mathematical model is

$$y_k[m] = s_k[n] + n_k[m] + a_k[m], \tag{12}$$

where s_k is the received signal when there is no system noise and no attack. The term n_k is the effective noise term which represents system noise in a mathematically equivalent formulation. Although several subsystems of an AV radar can generate different levels of noise, eventually their combined effect will appear as an added term, n_k, at the output of the ADC block (See Figure 3). The term a_k is the effective attack signal. Although one can consider so many different attack scenarios, eventually all attack signals will appear as an added, a_k, at the output of the ADC. This mathematical formulation allows us to study the effects of noise level, and cyberattack signal power level separately.

The motivation behind the proposed approach can be explained as follows: The AV radar system knows the random signature ρ, but does not know $a_k[m]$'s. On the other hand, the attacking agent may have full information about the AV radar system architecture, may even know chirp start/end times, but we assume that it does not know the random signature. Because of the rapid switching pattern of the AV radar, we assume that the attacking agent cannot determine the value of $\rho[k]$ while we are in the interval I_k. This is a very realistic assumption, because for the Texas Instruments radar used in this work, the transmitter is active for only 60 μs during every chirp, and for each chirp the slope is re-selected on a random basis. Identifying the nature of the received RF waveform, finding the slope of the chirp, and generating an intelligent counter attack signal in such a short period of time requires noticeable effort and resources. Basically, by the time that the attacking agent determines the slope, that chirp cycle will be over, and the next chirp cycle will begin with a new random slope.

Our cyberattack Detection DSP algorithm is presented in Algorithm 1. The main motivation behind this specific detector design is the following: For up-chirps, the complex beat signal will have only positive frequency components in its FFT decomposition, whereas for down-chirps, we will have only negative frequency components. In reality, because of system noise, we will always observe both negative and positive frequency components. However, for up-chirps, the average power at negative frequencies should be small, and for down-chirps, the average power at negative frequencies of the reversed signal should be small. By comparing this with the average power of system noise, we can design a detector. Of course, the threshold selection, and associated false alarm rates are important design parameters. These will be discussed in the following sections using real data.

In Line 3 of the Algorithm 1, we have $y_k \leftarrow y_k - \frac{1}{N_s}\sum_{m=0}^{N_s-1} y_k[m]$, which is a vector expression and should be interpreted as subtracting the constant $\frac{1}{N_s}\sum_{m=0}^{N_s-1} y_k[m]$ from each component of y_k simultaneously. In Line 4, $\rho[k]$ is value of the random signature function at k, and $\rho[k] = 1$ means positive ramp, and 0 means negative ramp is used for the k^{th} chirp. The $reverse(\cdot)$ means reversal in time domain, that is,

$$\text{if } z_k = reverse(y_k), \text{ then } z_k[m] = y_k[N_s - 1 - m] \text{ for } m = 0, \cdots, N_s - 1. \tag{13}$$

For up-chirps, we compute the average power at negative frequencies, and for down-chirps we first do time reversal and then compute the average power at negative frequencies. For a given one dimensional signal $y[m]$ defined for $m = 0, \cdots, N_s - 1$, the $Y = FFT(y)$ is defined as

$$Y[m] = \sum_{m_1=0}^{N_s-1} y[m_1] e^{-j2\pi\frac{mm_1}{N_s}}, \quad m = 0, \cdots, N_s - 1, \tag{14}$$

and the average power at negative frequencies is defined as

$$P_- = \frac{1}{N_s/2} \sum_{m=N_s/2}^{N_s-1} |Y[m]|^2 \tag{15}$$

Maximum of these average power values, P, are compared with $2\sigma_n^2$, where σ_n^2 is the average power of the system noise. Basically, a P value below $2\sigma_n^2$ does not trigger an alarm, however larger P values will be considered as suspicious, and the cyberattack alarm will be triggered.

Our AV sensor Radar DSP algorithm is presented in Algorithm 2. It is similar to a standard 2D FFT based range-velocity heatmap generation algorithm. As in the Detector DSP code, we start with DC offset removal, and then do time reversal for down-chirps. Then we define the complex matrix I, compute its 2D FFT, then do $fftshift$, followed by element-wise absolute value. The 2D FFT of I is denoted by $F = FFT(I)$, and is equal to

$$F[k,m] = \sum_{k_1=0}^{M-1}\sum_{m_1=0}^{N_s-1} I[k_1,m_1]e^{-j2\pi\frac{kk_1}{M}}e^{-j2\pi\frac{mm_1}{N_s}}, \quad k = 0,\cdots,M-1,\ m = 0,\cdots,N_s-1. \tag{16}$$

The $fftshift$ which appears in Line 6 of Algorithm 2 is defined as

$$fftshift\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \begin{bmatrix} D & C \\ B & A \end{bmatrix}, \tag{17}$$

provided that A, B, C, D are square matrices of same dimensions. Similary, the absolute value of a matrix is defined as the element-wise absolute value, namely for a given $p \times q$ matrix A, $B = |A|$ is defined as

$$B_{i,j} = |A_{i,j}|, \quad i = 0,\cdots,p-1,\ j = 0,\cdots,q-1. \tag{18}$$

In Line 8 of Algorithm 2, the notation $0_{p,q}$ means the $p \times q$ zero matrix.

Algorithm 1: Detector (a.k.a. Detection DSP)

```
Input: Frame data, $y_k[m]$, $k = 0,\cdots,M-1$, and $m = 0,\cdots,N_s-1$, and the random signature
       function $\rho$ for the given measurement cycle.
Result: ($dratio$, $decision$)

/* Initialize maximum negative frequency power to zero                */
$P \leftarrow 0$

/* DC offset removal and time reversal                                */
for $k = 0$ to $M-1$ do
    $y_k \leftarrow y_k - \frac{1}{N_s}\sum_{m=0}^{N_s-1} y_k[m]$
    if $\rho[k]$ is 0 then
        $y_k \leftarrow reverse(y_k)$

/* Loop through all chirps                                            */
for $k = 0$ to $M-1$ do
    $Y_k \leftarrow \mathrm{FFT}(y_k)/\sqrt{N_s}$
    $P_{k-} \leftarrow 2\left(\sum_{m=N_s/2}^{N_s-1}|Y_k[m]|^2\right)/N_s$
    $P \leftarrow \max\{P, P_{k-}\}$

/* Threshold comparison                                               */
$dratio = P/\sigma_n^2$
if $dratio < 2$ then
    $decision \leftarrow$ "NO", that is, no attack detected
else
    $decision \leftarrow$ "YES", that is, an attack detected
return ($dratio$, $decision$)
```

Algorithm 2: AV Sensor (a.k.a. Radar DSP)

```
Input: Frame data, $y_k[m]$, $k = 0, \cdots, M-1$, and $m = 0, \cdots, N_s - 1$, and the random signature
       function $\rho$ for the given measurement cycle.
Result: Range-velocity heatmap as a 2D real matrix or as an image.

/* DC offset removal and time reversal                                        */
for $k = 0$ to $M - 1$ do
    $y_k \leftarrow y_k - \frac{1}{N_s} \sum_{m=0}^{N_s-1} y_k[m]$
    if $\rho[k]$ is 0 then
        $y_k \leftarrow reverse(y_k)$

/* Define complex valued matrix                                                */
Let $I[k,m] \leftarrow y_k[m]$, for $k = 0, \cdots, M-1$, and $m = 0, \cdots, N_s - 1$

/* 2D FFT                                                                      */
Let $F \leftarrow FFT(I)$; then $F \leftarrow fftshift(F)$; then $F \leftarrow |F|$ that is, element-wise absolute value

Decompose $F$ into columns as $F = [F_0, \cdots, F_{N_s-1}]$

Let $F = F - [0_{M,N_s/2} \vdots F_{N_s/2-1} \cdots F_0]$

Delete the columns $m = 0, \cdots, N_s/2 - 1$ of the resulting $F$ matrix

Optional: Normalization, rotation, apply a colormap to $F$, etc.

return $F$
```

4. Experimental Setup and Measurements

In this section, we will briefly describe our experimental setup, and the equipment used for testing. As hardware, we have two 77 GHz Texas Instruments radars, a Lattice FPGA board for streaming the received data to a host PC, a host PC for recording radar data, and interface cables. The list of equipment used in our experiments are shown in Table 1. We have also two different software packages from Texas Instruments as shown in Table 2.

Table 1. List of equipment.

Model Number	Details
AWR1642BOOST	76-GHz to 81-GHz automotive radar (Texas Instruments AWR1642 chip based)
AWR1642BOOST-ODS	76-GHz to 81-GHz automotive radar (Texas Instruments AWR1642 chip based, wide field of view antenna)
DCA1000 EVM	FPGA board for real-time data-capture (Lattice LFE5UM-85F-8BG381I chip based)
HostPC	Laptop, 16GB RAM, i7-8750H CPU @ 2.20 GHz, 64-bit Windows 10, NVIDIA GeForce GTX 1050 Ti graphics accelerator
Other	Two 5 V power supplies, two USB 2.0 cables (for configuration), and a Gigabit Ethernet cable (for high speed data)

Table 2. List of software.

Name	Details
TI mmWave Studio	Version 2.1.0.0
TI mmWave Visualizer	Version 3.3.0

Our experimental setup shown in Figure 5 consists of two 77GHz radars. The radar on the left is AWR1642BOOST and is used to generate the attack signal. The radar on the right is AWR1642BOOST-ODS with wider field of view antenna, and is used as the AV sensor. This second radar receives the signals generated by its own transmitter, as well as the attack signals generated by the first radar. The mmWave Studio is used for the configuration of the radar used as the AV sensor (i.e., AWR1642BOOST-ODS), and the FPGA board DCA1000 EVM. The FPGA board is used for streaming the measured radar data to the host PC over a Gigabit Ethernet connection. Although recording can be done using the mmWave Studio software, we used our own Python programs for more flexible recording and real-time visualization options. The mmWave Visualizer software is used for operating the radar AWR1642BOOST which is used to generate attack signals. Both TI software packages, as well as our Python programs for recording and visualization run on the same host PC. In the following, the radars AWR1642BOOST and AWR1642BOOST-ODS are called the attack and the AV radars respectively.

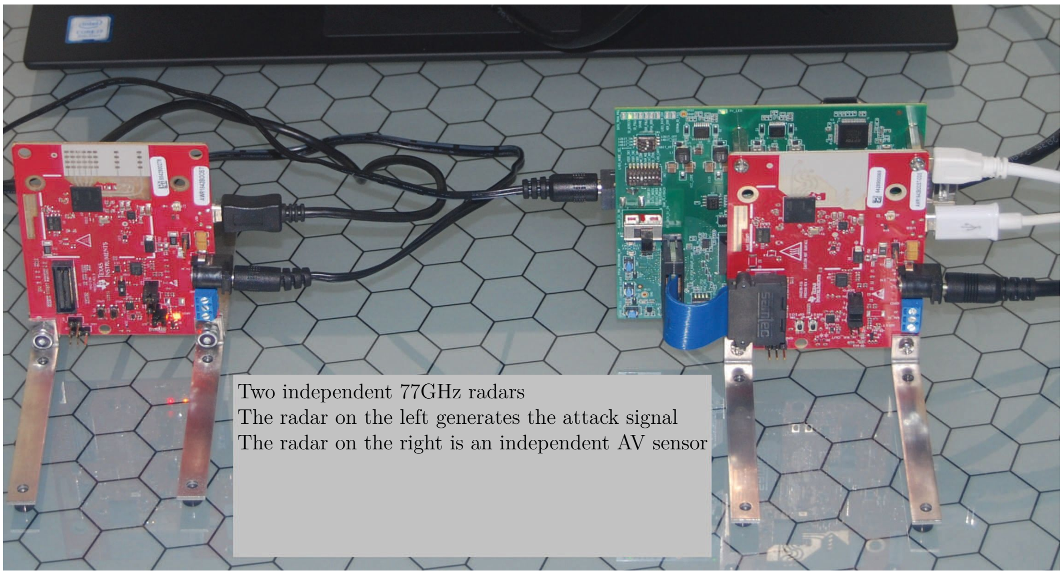

Two independent 77GHz radars
The radar on the left generates the attack signal
The radar on the right is an independent AV sensor

Figure 5. Our experimental setup consists of two 77 GHz radars. The radar on the left is used to generate the attack signal. The radar on the right is used as the AV radar sensor, and receives reflections of its own transmitted signal, as well as the attack signals generated by other radar.

We have done various field tests using the setup shown in Figure 6. Basically we have recorded data for cars driving in front of the radar, as well as pedestrians walking in front of the radar.

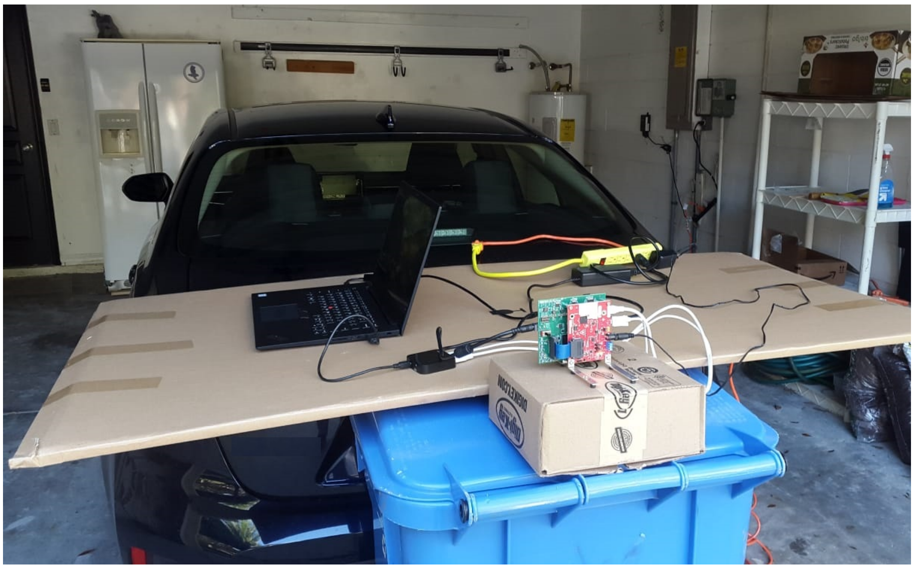


Figure 6. A field experiment using the AV radar sensor.

To be able to analyze different factors, we have conducted three different types of measurements as shown in Table 3.

Table 3. List measurement types.

Type	Details
Type I: Noise only	Attack Radar TX = Off ; AV Radar TX = Off, RX = On
Type II: Normal mode	Attack Radar TX = Off ; AV Radar TX = On, RX = On
Type III: Attack mode	Attack Radar TX = On ; AV Radar TX = Off, RX = On

For the first type of measurements, we turn off all transmitters and record only the system noise observed on the receive signal path of the AV radar. This will enable us to understand more about the nature of measurement noise. But more important than that, we can use amplified versions of the noise signal to generate different test data sets. For the second type of measurements, the attack radar is off, and the AV radar is used to record real data for various test cases. These test cases include a vehicle moving in front of the AV radar, and a pedestrian walking in front of the AV radar. Real data for both moving towards and away from the AV radar has been recorded, and this will also be quite useful to generate various test data sets. For the third and final type of measurements, the attack radar is on, AV radar's transmitter is off, but its receiver is on. This will enable us to record the effect of the attack signal only. Once we have these recordings, we can continuously scale both noise, and the attack signal in very small steps, and generate a very large number of test data sets for the proposed algorithms.

Basically, instead of positioning the attack radar in hundreds of different positions, we consider only a couple of different positions and record the effect of the attack radar for these cases only. But later we scale the effective attack signal, a_k, to generate hundreds of test data sets for varying levels of attack signal power. This kind of approach makes the radar experiments more controllable.

In a similar fashion, instead of driving the test vehicle in hundreds of different tracks to generate hundreds of different SNR levels, we consider only a couple of different tracks and record the received data for these cases only. But later we scale the effective noise signal, n_k, to generate hundreds of test data sets for varying levels of SNR. This kind of approach also makes the radar experiments more controllable.

Basically, we use the following equation to generate various test cases

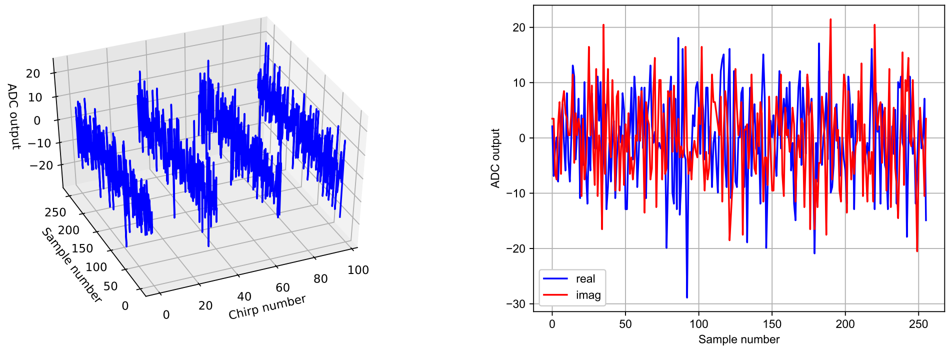
$$y_k = s_k + c_1 n_k + c_2 a_k, \tag{19}$$

where s_k is a signal recorded in a Type II measurement, n_k is a signal recorded in a Type I measurement, and a_k is a signal recorded in a Type III measurement. The constants c_1, c_2 are positive real numbers, and are used to generate test data sets with varying SNR, and cyberattack signal power levels. In the following sections, performance of the cyberattack Detection DSP algorithm, as well as the Radar DSP algorithm are analyzed by using these test data sets. The advantage of this approach is the fine control that we have on SNR, and cyberattack signal power levels.

5. Analysis of System Noise

In this section, we analyze statistical properties of system noise using a Type I measurement data. Basically, the analysis presented in this section shows that we can model the system noise as white Gaussian noise (WGN). Hence, instead of using a limited set of recorded noise data, we can generate synthetic noise data and do more realistic performance tests for the proposed detector.

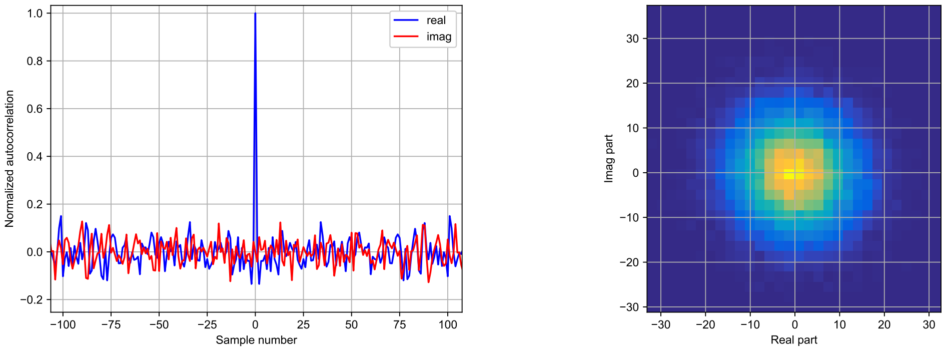
In Figure 7a, the real part of the measurement noise is shown in discrete time domain. There are basically $M = 128$ chirps, but only 4 of them are shown. For each chirp, we have a total of $N_s = 256$ samples. In Figure 7b, we have real and imaginary parts of system noise is shown for a single chirp.



(**a**) (**b**)

Figure 7. (**a**) Real part of the system noise for 4 chirps. (**b**) Real and imaginary parts of the system noise for a single chirp.

In Figure 8a, normalized autocorrelation of the system noise is shown. The autocorrelation is computed using the unbiased method, and the figure shows that the noise is quite white (Here we silently assume ergodicity of the noise process). In Figure 8b, we study the correlation between the real and imaginary part of system noise using actual data. The 2D histogram has a reasonably smooth circular shape, and the estimated correlation coefficient is less than 0.05 in absolute value. Finally, in Figure 9, we have the empirical PDFs of real and imaginary parts of system noise, and the fitted normal distributions with parameters estimated from real data.



(**a**) (**b**)

Figure 8. (**a**) Normalized autocorrelation of the system noise using the unbiased method. (**b**) 2D histogram of the system noise shown in Parula colormap. The empirical cross correlation between real and imaginary parts is estimated as ≤ 0.05 in absolute value.

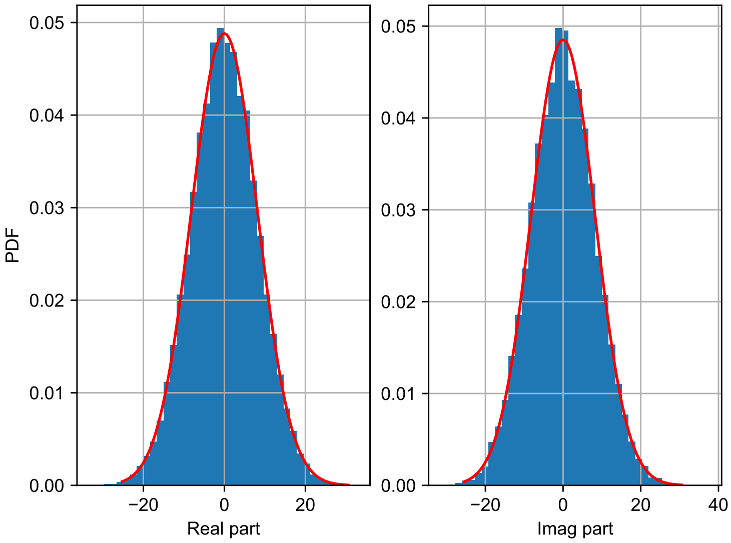


Figure 9. Empirical PDFs of real and imaginary parts of the system noise (Blue), and matching normal distributions (Red).

The analysis presented above shows that the system noise can be viewed as a white Gaussian noise. Based on this observation, we can generate different test data sets: (1) Using recorded system noise and scaled versions of it, or (2) using just pseudo random number generators without using a recorded noise data.

6. Analysis of the cyberattack Detection DSP

In this section, we will demonstrate the effectiveness of the proposed cyberattack Detection DSP, and summarize our test case generation methodology. Basically, to generate a test case, we will first choose a radar data, s_k, which is recorded with no extra noise, or attack signal source. Then we add a scaled version of system noise, and then a scaled version of a recorded attack signal. As summarized in Table 3,

(1) For a recorded system noise, n_k, all transmitters are off (a.k.a. Type I measurement), so that we capture only the system noise,

(2) For a recorded radar data, s_k, the Attack Radar TX is off (a.k.a. Type II measurement), so we capture only the signal reflected from a target (A car or a pedestrian),
(3) For a recorded attack signal, a_k, Attack Radar TX is on, but AV Radar TX is off (a.k.a. Type III measurement). In this case, we capture only the effect of the attack signal.

Then we use $y_k = s_k + c_1 n_k + c_2 a_k$ to generate multiple test cases. This weighted addition is done after all measurements are completed, and allows us to play with relative strengths of noise and attacks signals easily. In summary, we use recorded radar data s_k, recorded or synthetic noise, n_k, and recorded attack signal, a_k, and do the weighted summation to generate test cases.

Our test cases can be grouped under 5 different headings as shown in Table 4.

Table 4. Test cases.

Attack Scenario	**Details**
eWGN	Attack signal is white noise
eTX2	Attack signal is the signal generated by the second radar unit
eTX2R	Attack signal is the signal generated by the second radar unit with random frequency slope selection
eTX1	Attacker has full VCO sync information
eTX1R	Attacker has full VCO sync information, and selects the frequency slope randomly

Basically, eWGN corresponds to an experiment where the attacker generating white Gaussian noise. The case eTX2 corresponds to an experiment where Attack Radar TX is generating successive chirp signals to attack/confuse the AV Radar. The case eTX2R is basically the same as eTX2 with random frequency slope selection on the Attack Radar. Namely, the Attack Radar is randomly choosing between positive and negative frequency slopes. For the eTX2 and eTX2R cases, VCO's of the AV Radar and the Attack Radar are not synchronized. To test VCO synchronized attacks, we have two more test cases, eTX1 and eTX1R. The eTX1 case, corresponds to an experiment where the Attack Radar is generating a TX signal for a different environment with possibly fake targets. The eTX1R is basically the same as eTX1 with random frequency slope selection on the Attack Radar.

The attack types WGN, eTX2, and eTX2R look more standard attack types, but eTX1 and eTX1R are smarter attacks where Attack Radar TX is generating a signal which will cause the AV Radar to sense a different environment with possibly fake targets. However, the random signature generator block implemented in the proposed AV Radar system detects such attacks.

6.1. Analysis of False Alarms

In this subsection, we present detector statistics when there is no attack, and use these to *estimate* the false alarm rate. Test cases presented in Table 4 will be analyzed in the following subsection.

In Figure 10, we have the PDF of the detector output when there is no attack signal. To generate this PDF, we have used a recorded data set consisting of 1000 frames. Basically a car was driven in front of the radar, no attack signal was generated, and the received radar data was recorded. Recall that, the algorithm used in the Detection DSP block is presented in Algorithm 1, and outputs a real value, *dratio*, and a boolean value *decision*. The PDF presented in Figure 10 is for the real value, *dratio*. As seen in Figure 10, *dratio* is always below 2.

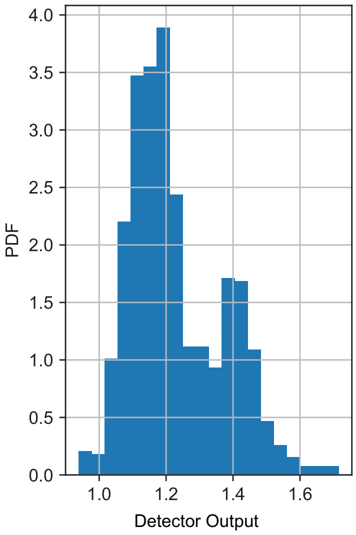


Figure 10. Detector output's emprical PDF.

To estimate the false alarm rate, we first model this empirical PDF as a normal distribution. The best fitted normal distribution seems to be $N(1.23, 0.14)$. Under the assumption that the detector output is a normal distribution, the probability of detector output, *dratio*, exceeding the threshold 2 can be approximated as

$$Prob\left\{1.23 + 0.14x \geq 2\right\}, \tag{20}$$

where x is $N(0, 1)$ random variable. This is basically the probability of false alarm. Using standard identities for the CDF, Φ, and the error function, erf, we get

$$\Phi((2 - 1.23)/0.14) = (1 + erf((2 - 1.23)/0.14/\sqrt{2})) \approx 1 - 2 \times 10^{-8}. \tag{21}$$

Therefore, the false alarm rate is estimated as well below 10^{-6}.

Authors tried multiple other recorded data sets but, despite all attempts, were unable to generate even a single false alarm.

6.2. eWGN Cyberattack Test Case

In this test case, the attack signal a_k is a randomly selected frame of a recorded system noise (Type I recording). Note that, we multiply this a_k with a coefficient c_2, and increase or decrease the strength of the attack signal for a more a detailed analysis.

The radar data s_k is a recorded receive signal when there is a moving target, but no attack signal (Type II recording). This is basically obtained from a randomly selected frame of a recording when a vehicle is being driven in front of the AV radar sensor.

The noise signal n_k is another randomly selected frame of a recorded system noise (Type I recording). Note that, we multiply this n_k with a coefficient c_1, and increase or decrease the strength of the noise signal to obtain different SNR levels.

As seen in Figure 11b, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. Plots in Figure 11 also contain experimental results from 100 different synthetically generated noise signals, n_k. For each SNR level, we have a min and max of detector output computed and shown in same color. Basically, upper and lower envelopes seem to be very close to each other, indicating little variation with respect to the actual values in the noise sequence. The same analysis is done for all of the other cases discussed in this section.

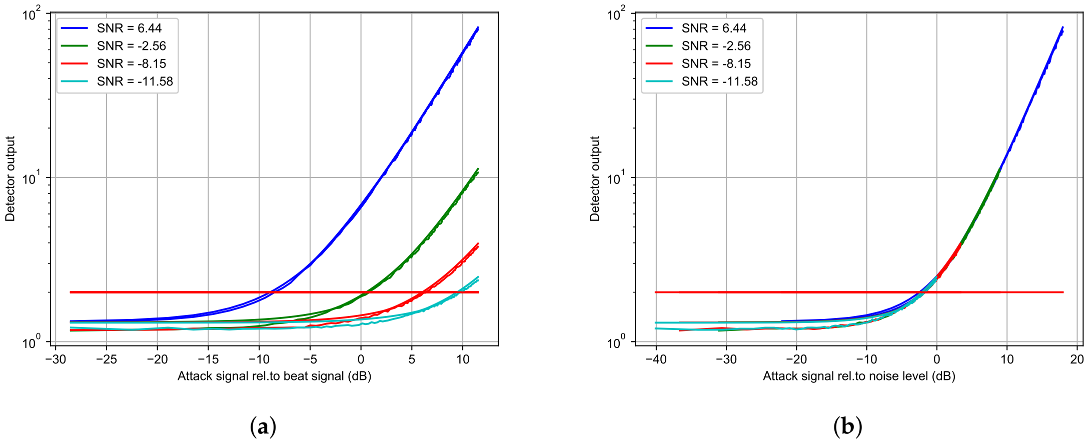


Figure 11. Test case eWGN. The horizontal orange colored line is the detection threshold. In (**a**) with respect to the beat signal level, and in (**b**) with respect to the noise level.

Figure 11a presents these from a different perspective: For a high SNR case, attacks with power levels well below the beat signal power level will be detected easily. As SNR level degrades, and noise level increases, attack signals need more power to trigger an alarm. This is related to a very important question: What is the “maximum” harmful effect that can be induced by an attack signal which is just below the detection threshold. How it effects target detection, and distance/velocity estimation performance? These are all addressed in the next section.

6.3. eTX2 Cyberattack Test Case

This case is similar to eWGN, but the attack signal is different. In this test case, the attack signal a_k is a randomly selected frame of a recorded attack signal (Type III recording). During this recording, Attack Radar’s TX was on, but the AV Radar’s TX was off. Using the AV Radar’s RX system, we have recorded just the effect of the attack signal. As in the previous case, we also scale the noise, and attack signals with different values for a more detailed analysis.

As seen in Figure 12b, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. The detector performance plots given in Figure 12a show that, as SNR get betters, detection of attacks will be easier.

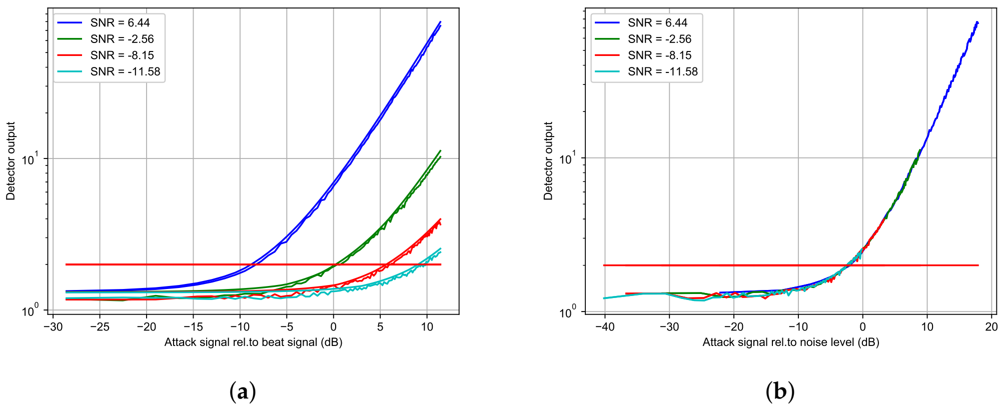
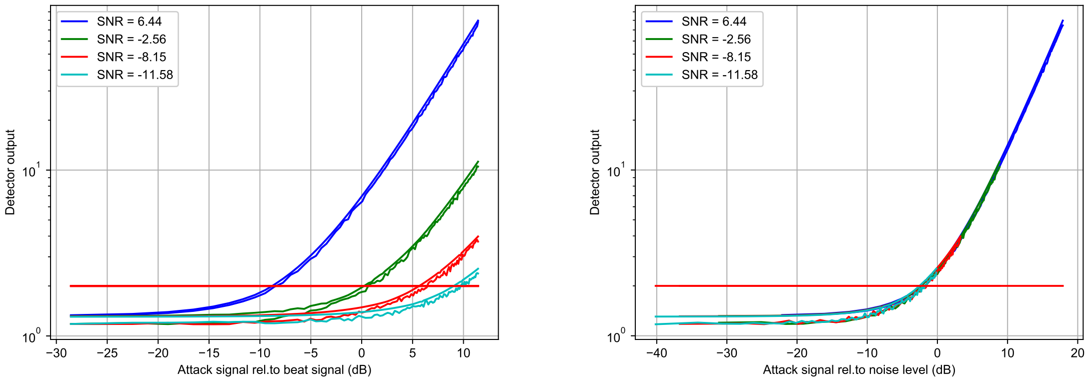


Figure 12. Test case eTX2. The horizontal orange colored line is the detection threshold. In (**a**) with respect to the beat signal level, and in (**b**) with respect to the noise level.

6.4. eTX2R Cyberattack Test Case

This case is very similar to the eTX2 case, but the Attack Radar is generating chirp signals with randomly selected slopes. Namely, randomly choosing a positive or negative frequency slope for each and every chirp in a frame.

As seen in Figure 13b, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. The detector performance plots given in Figure 13a show that, as SNR get betters, detection of attacks will be easier.



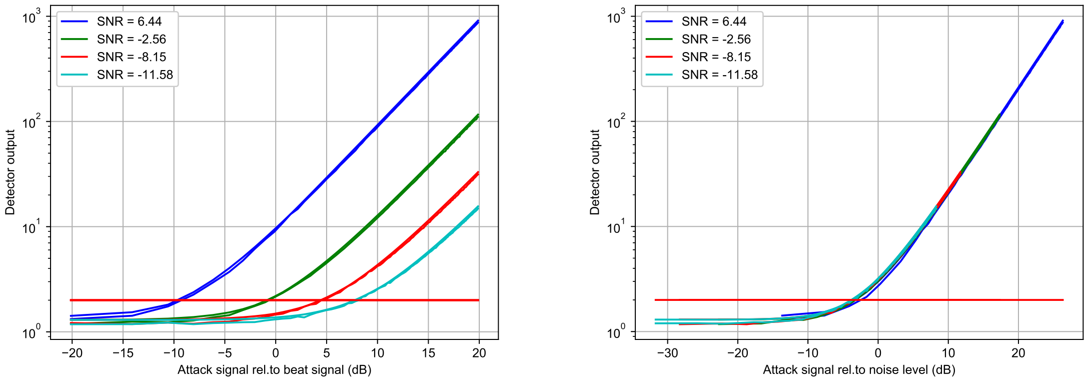
(**a**) (**b**)

Figure 13. Test case eTX2R. The horizontal orange colored line is the detection threshold. In (**a**) with respect to the beat signal level, and in (**b**) with respect to the noise level.

6.5. eTX1 Cyberattack Test Case

This case is similar to eWGN, but the attack signal is different. In this test case, the attack signal a_k is a randomly selected frame of a recorded signal in a Type II recording. This corresponds to a case where the Attack radar and the AV radar has full VCO synchronization. Practically, this is not easy to achieve, but represents a more challenging and smarter attack test case.

As seen in Figure 14b, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. The detector performance plots given in Figure 14a show that, as SNR get betters, detection of attacks will be easier.



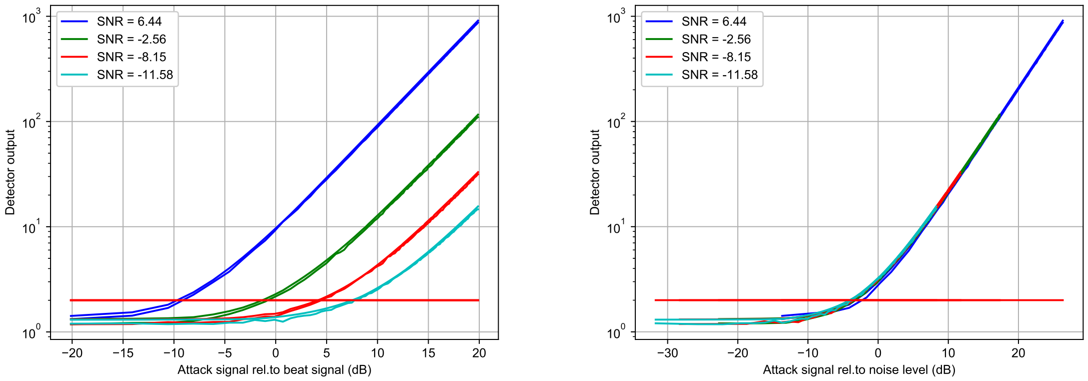
(**a**) (**b**)

Figure 14. Test case eTX1. The horizontal orange colored line is the detection threshold. In (**a**) with respect to the beat signal level, and in (**b**) with respect to the noise level.

6.6. eTX1R Cyberattack Test Case

This case is very similar to the eTX1 case, but the Attack Radar is generating chirp signals with randomly selected slopes. Namely, randomly choosing a positive or negative frequency slope for each and every chirp in a frame.

As seen in Figure 15b, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. The detector performance plots given in Figure 15a show that, as SNR get betters, detection of attacks will be easier.



(**a**) (**b**)

Figure 15. Test case eTX1R. The horizontal orange colored line is the detection threshold. In (**a**) with respect to the beat signal level, and in (**b**) with respect to the noise level.

6.7. Detector Performance When There Is No Target

In this final subsection, we consider the case when there is no target. Namely, the AV radar receives just a noise like signal. To test the detector's performance, we use randomly selected frames of a noise recording as s_k, and n_k. As the attack signal, we used the signal used in the eTX2 case.

As seen in Figure 16, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. However, compared to the previous cases, a slightly more powerful attack signal is required to trigger the alarm.

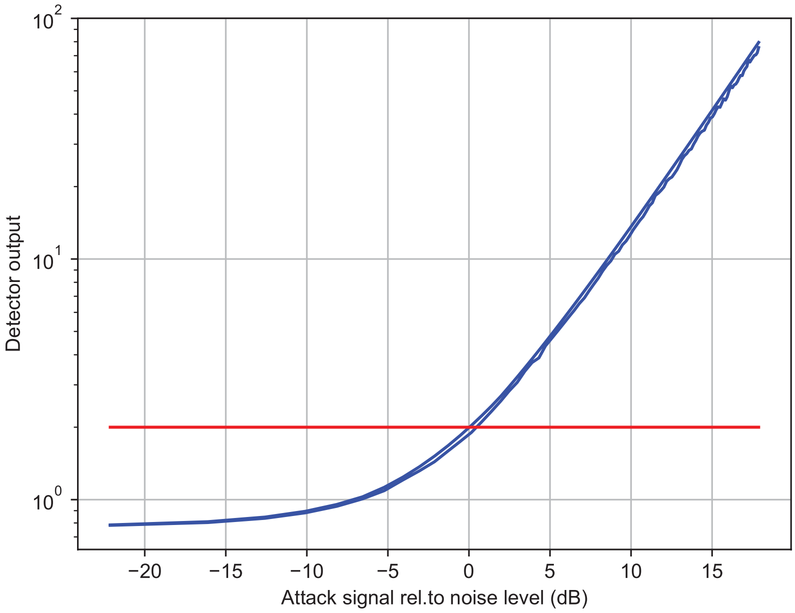


Figure 16. No target case. The horizontal orange colored line is the detection threshold.

7. Radar Performance under Cyberattacks: Analysis of Radar DSP

We have seen that, when the attack signal becomes comparable to noise level, the detector output will exceed the threshold 2 and cyberattack alarm will be triggered. But what could be the worst case effect of undetected cyberattack signals? Namely, cyberattack signals which are relatively strong, but just below the detection threshold. In this section, we present 3 example cases (All eTX2 type). For each case, we have the radar DSP output when there is no attack, and when there is an attack which is just below the detection threshold. As seen in Figure 17, we see little difference in the radar DSP output, and in all cases the approaching object will be detected because of the bright yellow dot.

In these figures, the center line is the static background, and the bright yellow dot corresponds to an approaching vehicle. Basically, an attack signal just below the detection threshold does have an effect on the 2D range/velocity heat-map, and that's why the left and right images are different. Furthermore, the difference seem to be more noticeable for poor SNR cases. But this small difference seem to have no noticeable effect for approaching object detection. Because, for all cases presented in Figure 17, an approaching object is clearly noticeable because of the bright yellow dot. Difference also seems to be minor for range/velocity estimation, because the bright yellow dot seems to be almost at the same location for all cases.

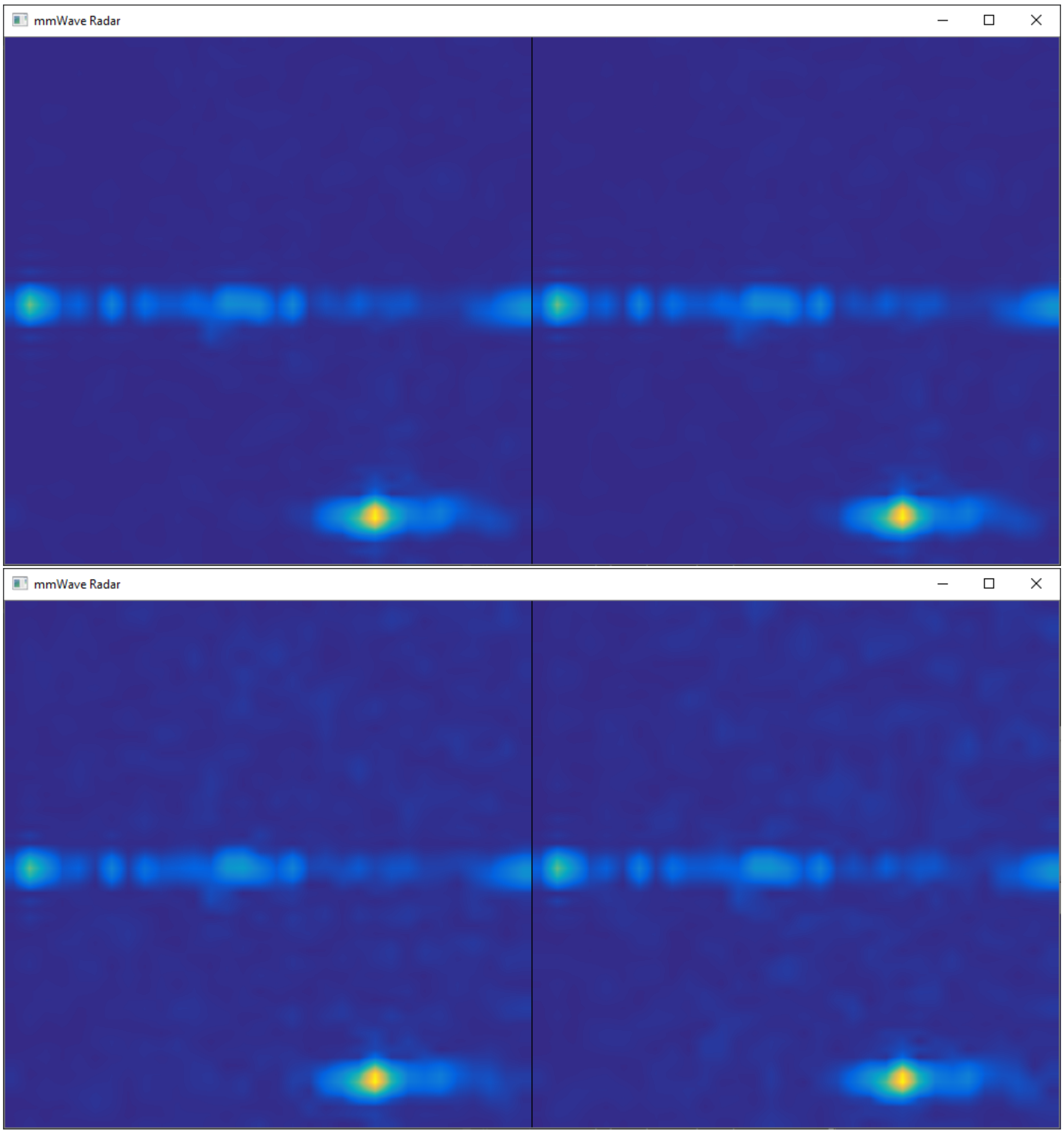


Figure 17. *Cont.*

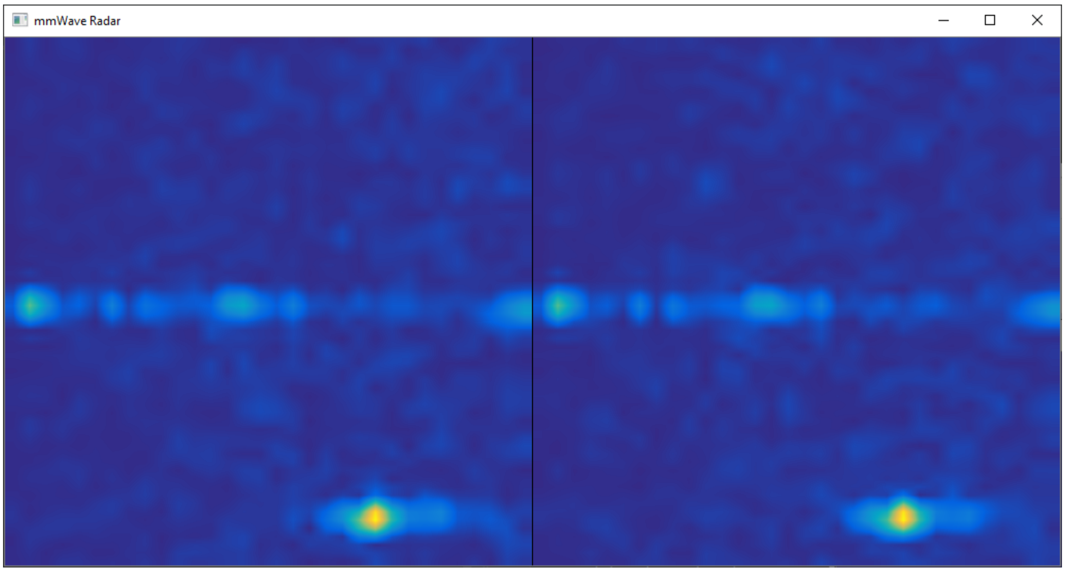


Figure 17. Radar DSP output for SNR = 6.44, −0.46, and −5.76. On the left, no attack, and on the right an attack just below the detection threshold.

8. Conclusions

In this paper, we have presented a cyberattack detection DSP algorithm, and an attack resilient radar DSP algorithm for 77 GHz AV radars. The proposed algorithm seem to have a very low false alarm rate, and for cyberattacks with power comparable to or larger than the noise level, all attacks seem to be detected. Finally, attacks which are just below the detection threshold and hence missed seem to have no noticeable effect on approaching target detection, and range/velocity estimation. All of these tests are done using real data collected by using the TI AWR series 77 GHz radars, and the real-time framework developed by the authors of References [19,21]. To demonstrate the effectiveness of the proposed algorithms, several attack scenarios are carefully analyzed. Both of the proposed DSP algorithms can easily be realized as radar firmware, and executed on the radar DSP processor. Finally, the proposed system is based on a standard FMCW radar RF front-end, and does not require fancy and more expensive RF components.

Author Contributions: Both authors (O.T., S.A.) contributed to Conceptualization, Investigation, and Methodology. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Florida Polytechnic University Startup Fund, and the National Science Foundation Grant No. CNS-1919855.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kala, R. *On-Road Intelligent Vehicles, Motion Planning for Intelligent Transportation Systems*; Butterworth-Heinemann: Oxford, UK, 2016. doi:10.1016/C2015-0-00389-6. [CrossRef]
2. Jimenez, F. *Intelligent Vehicles, Enabling Technologies and Future Developments*; Butterworth-Heinemann: Oxford, UK, 2018. doi:10.1016/C2016-0-04123-2. [CrossRef]
3. Patole, S.M.; Torlak, M.; Wang, D.; Ali, M. Automotive radars: A review of signal processing techniques. *IEEE Signal Proc. Mag.* **2017**, *34*, 22–35. doi:10.1109/MSP.2016.2628914. [CrossRef]
4. Ramasubramanian, K.; Ramaiah, K.; Aginskiy, A. Moving from legacy 24 GHz to state-of-the-art 77 GHz radar. *ATZelektronik Worldw.* **2018**, *13*, doi:10.1007/s38314-018-0029-6. Available online: http://www.ti.com/lit/wp/spry312/spry312.pdf (accessed on 22 March 2020). [CrossRef]
5. Bhat, C. *Cybersecurity Challenges and Pathways in the Context of Connected Vehicle Systems*; Technical Report 134; Data-Supported Transportation Operations & Planning Center (D-STOP): Austin, TX, USA, 2018. Available online: https://ctr.utexas.edu/wp-content/uploads/134.pdf (accessed on 22 March 2020).

6. Alland, S.; Stark, W.; Ali, M.; Hegde, M. Interference in Automotive Radar Systems: Characteristics, Mitigation Techniques, and Current and Future Research. *IEEE Signal Proc. Mag.* **2019**, *36*, 45–59. doi:10.1109/MSP.2019.2908214. [CrossRef]
7. Ye, N.; Farley, T. A scientific approach to cyberattack detection. *Computer* **2005**, *38*, 55–61. doi:10.1109/MC.2005.358. [CrossRef]
8. Pham, V.; Nguyen, N.; Li, J.; Hass, J.; Chen, Y.; Dang, T. MTSAD: Multivariate Time Series Abnormality Detection and Visualization. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019. doi:10.1109/BigData47090.2019.9006559. [CrossRef]
9. Xu, L.; Liu, H.; Zhou, S.; Liu, J.; Yan, J. Colocated MIMO Radar Waveform Design Against Repeat Radar Jammers. In Proceedings of the 2018 International Conference on Radar (RADAR), Brisbane, Australia, 27–31 August 2018. doi:10.1109/BigData47090.2019.9006559. [CrossRef]
10. Chen, H.; Himed, B. Analyzing and improving MIMO radar detection performance in the presence of cybersecurity attacks. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Pacific Grove, CA, USA, 6–9 November 2016. doi:10.1109/RADAR.2016.7485177. [CrossRef]
11. Yang, C.; Feng, L.; Zhang, H.; He, S.; Shi, Z. A Novel Data Fusion Algorithm to Combat False Data Injection Attacks in Networked Radar Systems. *IEEE Trans. Signal Inf. Process. Over Netw.* **2018**, *4*, 125–136. doi:10.1109/TSIPN.2018.2790361. [CrossRef]
12. Kordestani, M.; Chibakhsh, A.; Saif, M. A Control Oriented Cyber-Secure Strategy Based on Multiple Sensor Fusion. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019. doi:10.1109/SMC.2019.8914241. [CrossRef]
13. Ye, N.; Farley, T. Resilient Tracking Control of Networked Control Systems Under Cyber Attacks. *IEEE Trans. Cybern.* **2019**, 1–13. doi:10.1109/TCYB.2019.2948427. [CrossRef]
14. Mo, Y.; Sinopoli, B. False Data Injection Attacks in Control Systems. First Workshop on Secure Control Systems, CPS Week. 2010. Available online: https://www.researchgate.net/publication/228859026_False_data_injection_attacks_in_control_systems (accessed on 22 March 2020).
15. Shoukry, Y.; Martin, P.; Yona, Y.; Diggavi, S.; Srivastava, M. Attack Resilience and Recovery using Physical Challenge Response Authentication for Active Sensors Under Integrity Attacks. *arXiv* **2016**, arXiv:1605.02062v2. Available online: https://arxiv.org/pdf/1605.02062 (accessed on 22 March 2020).
16. Dutta, R.G.; Guo, X.; Zhang, T.; Kwiat, K.; Kamhoua, C.; Njilla, L.; Jin, Y. Estimation of safe sensor measurements of autonomous system under attack. In Proceedings of the 54th ACM/EDAC/IEEE Design Automation Conference, Austin, TX, USA, 19–23 June 2017. doi:10.1145/3061639.3062241. [CrossRef]
17. Kapoor, P.; Vora, A.; Kang, K.D. Detecting and Mitigating Spoofing Attack against an Automotive Radar. In Proceedings of the 2018 IEEE Vehicular Technology Conference, Chicago, IL, USA, 27–30 August 2018. doi:10.1109/VTCFall.2018.8690734. [CrossRef]
18. Toker, O.; Alsweiss, S.; Vargas, J.; Razdan, R. Design of an Automotive Radar Sensor Firmware Resilient to Cyberattacks. In Proceedings of the 2020 IEEE SoutheastCon, Raleigh, NC, USA, 11–15 March 2020.
19. Toker, O.; Alsweiss, S.; Abid, M. A Computer Vision Based Testbed for 77 GHz mmWave Radar Sensors. In Proceedings of the 2020 IEEE SoutheastCon, Raleigh, NC, USA, 11–15 March 2020.
20. Toker, O.; Alsweiss, S. mmWave Radar Based Approach for Pedestrian Identification in Autonomous Vehicles. In Proceedings of the 2020 IEEE SoutheastCon, Raleigh, NC, USA, 11–15 March 2020.
21. Toker, O.; Kuhn, B. A Python Based Testbed for Real-Time Testing and Visualization using TI's 77 GHz Automotive Radars. In Proceedings of the 2019 IEEE Vehicular Networking Conference, Los Angeles, CA, USA, 4–6 December 2019.
22. Brinkmann, M. Design and Implementation of Improved Nonlinearity Correction Algorithms for FMCW Radar Sensors. Master's Thesis, Florida Polytechnic University, Lakeland, FL, USA, 2019.
23. Toker, O.; Brinkmann, M. A Novel Nonlinearity Correction Algorithm for FMCW Radar Systems for Optimal Range Accuracy and Improved Multitarget Detection Capability. *Electronics* **2019**, *11*, 1290. doi:10.3390/electronics8111290. [CrossRef]
24. Iovescu, C.; Rao, S. The Fundamentals of Millimeter Wave Sensors. Texas Instruments Technical Document. 2017. Available online: http://www.ti.com/lit/wp/spyy005/spyy005.pdf (accessed on 22 March 2020).
25. Rao, S. Introduction to mmWave Sensing: FMCW Radars. Texas Instruments Training Document. 2017. Available online: https://training.ti.com/node/1139153 (accessed on 22 March 2020).

26. Ash, M.; Ritchie, M.; Chetty, K. On the Application of Digital Moving Target Indication Techniques to Short-Range FMCW Radar Data. *IEEE Sens. J.* **2018**, *18*, 4167–4175. doi:10.1109/TBCAS.2019.2908198. [CrossRef]
27. Park, J.; Hong, Y.; Lee, H.; Jang, C.; Yun, G.; Lee, H.; Yook, J. Noncontact RF Vital Sign Sensor for Continuous Monitoring of Driver Status. *IEEE Trans. Boimed. Circuits Syst.* **2019**, *13*, 493–502. doi:10.1109/TBCAS.2019.2908198. [CrossRef]
28. Aalizadeh, M.; Shaker, G.; Almeida, J.C.M.; Morita, P.P.; Safavi-Naeini, S. Remote Monitoring of Human Vital Signs Using mm-Wave FMCW Radar. *IEEE Access* **2019**, *7*, 54958–54968. doi:10.1109/ACCESS.2019.2912956. [CrossRef]
29. Toker, O. REAL-time 2D Radar Heatmap Demo Using the TI 77 GHz Radar. YouTube Video. Available online: https://www.youtube.com/watch?v=jNuBgl6KMCs (accessed on 22 March 2020).