Small Target Detection in a Radar Surveillance System using Contractive Autoencoders

SIMON WAGNER, Member, IEEE
WINFRIED JOHANNES
DENISA QOSJA
STEFAN BRÜGGENWIRTH, Senior Member, IEEE
Fraunhofer Institute for High Frequency Physics and Radar Techniques FHR

***Abstract*—With the rapid development of Unpiloted Aerial Vehicles (UAVs), also known as drones, in recent years, the need for surveillance systems that are able to detect drones has grown as well. Radar is technology with the potential to fulfill this task and several previous publications show examples of radar detection and classification schemes. The purpose of this paper is related to the detection scheme used in these radars. Most surveillance systems use a background subtraction and a threshold to detect targets. This threshold often depends on a model of the radar noise and the background, which is imperfect by nature. The approach presented here uses a data driven machine learning algorithm that is trained with measured background profiles of the radar and is applied afterwards to the given background for target detection. This scheme can in general be applied to any detection problem in a fixed area, but is shown here with examples from measurements of drones and persons. The results show that the chosen approach gives better detection rates for low false alarm rates with real data than background subtraction.**

***Index Terms*—Autoencoder, Drone Detection, Machine Learning, Radar Surveillance, Target Detection**

I. INTRODUCTION

The reliable detection and recognition of small targets, e.g. drones or persons, in heterogeneous clutter is still a major challenge for radar systems. One of the main applications, which drives research on this topic is airport security. During the last years, an increasing amount of airport closings due to drone incidents is observed [1]. Famous examples are the attacks on Gatwick and Heathrow airports in London, UK, in December 2018 and January 2019. In May 2019, Frankfurt Airport in Germany was shut down for an hour because of a drone sighting. These and other examples can be found in [1] together with an overview on countermeasures and different sensors, including radar, to track and identify drones.

Because of this urgent need for a reliable detection and classification system for small targets, a lot of research was conducted over the last years on this topic. The used methods and scenarios show a large variety, e.g. passive radar [2], noise radar [3] or the detection of insect-like size nano-drones [4]. An overview of machine learning based methods for detection and classification can be found in [5], which also shows approaches from acoustics, optics and radio frequency identification. Hardware aspect to improve the sensitivity of radars to improve the detection of small targets can be found for example in [6] and [7]. The latter actually presents a system with a similar processing as the one used in this work. It should be mentioned that most of the radar research is focused on classification of drones, often by micro-Doppler signatures and kinematic features, e.g. [8], while the detection is assumed to be done beforehand.

The radar examples mentioned above have in common that they use a radar with a limited observation area and a rather long integration time. The approach presented here is limited to detection rather than classification, but uses a rotating surveillance radar with a mechanical staring antenna and the detection is based on a single snapshot, i.e. the range profile created by a single pulse without integration gain. This means that neighboring profiles are not used and targets are detected in each range profile individually. However, it is assumed that the background of the measurement is known to the radar and therefore, only changes must be detected.

The radar used in the work here is a surveillance radar at a fixed position and the natural background is assumed to be stationary with small variations due to wind or other effects. These variations of the clutter should be suppressed by the network, which is a *contractive autoencoder*. An autoencoder is designed to reproduce the input data at the output of the network and a so-called *code* is generated as an intermediate result, which ideally contains the entire information about the input signal [9]. Furthermore, the contractive autoencoder is designed to be robust against small variations in the input signal. The determination of the contractive term in the cost function is a trade-off between clutter and false alarm suppression.

Manuscript received XXXXX 00, 0000; revised XXXXX 00, 0000; accepted XXXXX 00, 0000.
This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. This work was supported in part by the NATO Communication and Information Agency (NCIA) under Grant PO 42008950. *(Corresponding author: S. Wagner).*

S. Wagner, Fraunhofer Institute for High Frequency Physics and Radar Techniques, FHR, Cognitive Radar, Fraunhoferstraße 20, 53343 Wachtberg, Germany (e-mail: simon.wagner@fhr.fraunhofer.de).

Mentions of supplemental materials and animal/human rights statements can be included here.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

0018-9251 © 2022 IEEE

Details about the network and the detection algorithm will be given in section IV.

The reference methods that are used in this work are mean and Gaussian background subtraction. For these methods, a mean background range profile created by the independently measured background range profiles averaged over time is used to subtract it from new data. In case of the Gaussian background subtraction, an additional window is calculated to further suppress areas with high variations in the background. This means that measurements of the observed area without any targets must be available. In this way, the targets must be detected against the noise floor that is created by the variations of the background in the radar data. However, over the last years, several machine learning and artificial intelligence methods have been presented for change detection [10]. Since these machine learning methods outperformed classical methods in many tasks, an approach based on neural networks is chosen in this paper.

Beside the mentioned results on change detection, machine learning and neural networks have also been used for different detection tasks in radar. For example, [11] trained a neural network to replace a Constant False Alarm Rate (CFAR) detector and showed results for simulations as well as real measurements. The difference to the work here is the used network architecture and the fact that we assume the background as known and thus perform change detection. The replacement of a CFAR algorithm by a neural network was also investigated by [12] from a theoretical point of view. They showed that the Neyman-Pearson Detector, i.e. the CFAR detector, can be approximated by a neural network if squared error cost function is used. Other work related to detection in the radar domain is mainly focused on imaging radar with so called single shot detectors, e.g. [13].

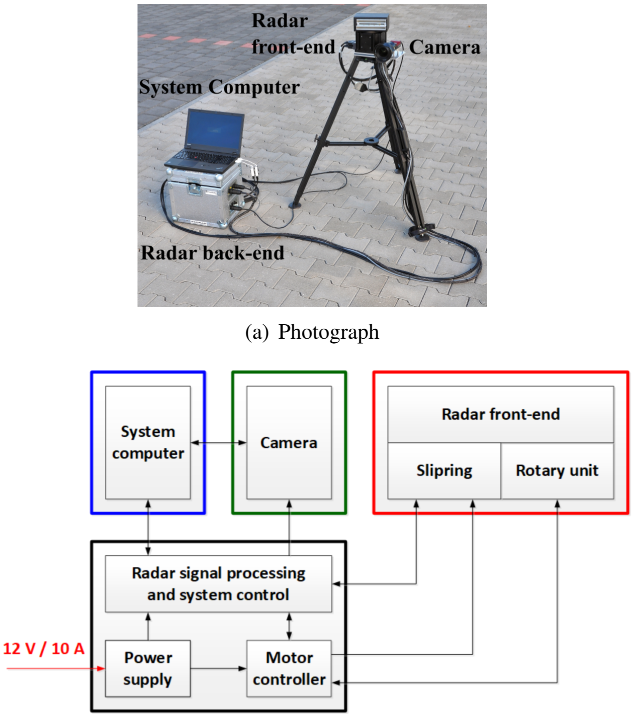
The method proposed here is a further development of a proof of concept approach presented in a previous paper of the authors [14]. The main changes compared to the early version are the following. A contractive term is included in the cost function and the targets in this paper are persons and drones with high fluctuations in the received echo. The detection in [14] was only shown for two corner reflectors with a constant RCS. Furthermore, two additional training steps called bias freezing and target only retraining are introduced here. The improvement in detection performance will be shown in section V by a comparison of the presented algorithm and a network as it was presented in [14].

The goal of the training process and the main contribution of this work is to design a network that captures the statistical properties of a given background and allows a reconstruction of the input data where high peaks due to background variations are suppressed. Therefore, the network should allow a more reliable detection of targets than the reference method. The evaluation of the detection scheme is done by simulations, as well as measurements. A description of the measured scenarios and the radar system is given in the following section.

II. Radar System and Measurements

A. Radar System

This section presents the Scanning Surveillance Radar System (SSRS). The system consists of a very compact and low weight rotating radar front-end in 94 GHz technology, a rotary unit, an optical camera, a system computer, a radar back-end in the compact and rugged PC/104 standard, a motor controller and a power supply. Figure 1 shows a photograph and a block diagram of the different elements of the system.



(a) Photograph

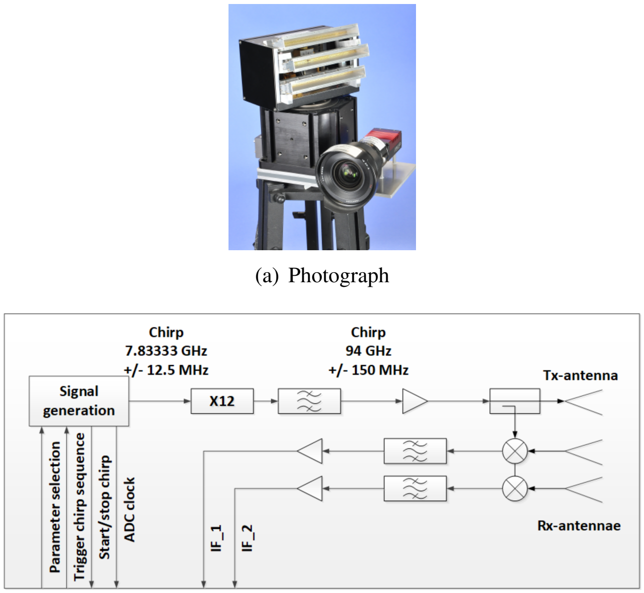
(b) Block diagram

Fig. 1. The Scanning Surveillance Radar System

The radar sensor, which is depicted in the block diagram of Figure 1 (b) in the red box, consists of a radar front-end, a slipring and a rotary unit. The slipring transfers the signals of the rotating front-end to the back-end (black box). Additionally, a camera (green box) and the system computer (blue box) are connected to the back-end. The back-end itself comprises the power supply with 12 V input voltage, the motor controller and a computer for digitizing and system control. The system computer is used to handle the data streams of the radar back-end and the camera, to control the whole system, to do the data processing and to visualize the processed data. Furthermore, the visualization of the data can be transferred via wireless network to up to 10 additional observers.

1. *Radar front-end*

The radar front-end is mounted on a rotary unit. Figure 2 shows a photograph and a block diagram of the radar front-end.



(a) Photograph

(b) Block diagram

Fig. 2. The front-end of the SSRS

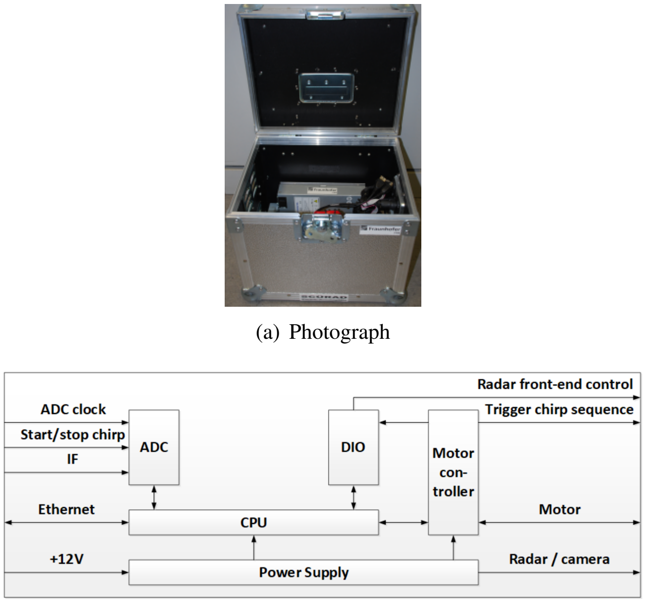
The signal generation unit of the radar front-end is based on a Direct Digital Synthesis (DDS) chip, which generates a highly linear chirp. The chirp bandwidth is 25 MHz with a center frequency of 7.83333 GHz. Additionally to the radar waveform, synchronization signals for the radar back-end are generated. The center frequency of 94 GHz with a bandwidth of 300 MHz is reached after frequency multiplication by 12. After a high power amplifier with an output power of 100 mW, the chirp is separated into the signal for the two highly sensitive mixers and for transmission via the Tx-antenna. The system has three slotted antennae with an aperture of 16 cm and a narrow beamwidth of 1.3° each – one for transmitting and two for receiving. The transmit chirp and the two received chirps are mixed and the resulting intermediate frequencies are filtered and amplified. A slipring transfers these signals and some synchronization signals to analog to digital converters located in the radar back-end. Table I shows the technical specification of the radar front-end.

TABLE I
Specifications of the radar front-end

Radar type	FMCW
Operating frequency	94 GHz
Maximum Radar bandwidth	1 GHz
Maximum range resolution	15 cm
Output power	100 mW
Slotted antenna	1.3° x 13°
Physical dimensions	185 × 145 × 212 mm^3
Weight	8551.2 g

2. *Radar back-end*

The radar back-end is necessary to accomplish the signal processing of the intermediate frequency and the system control. Figure 3 shows a photograph and a block diagram of the radar back-end.



(a) Photograph

(b) Block diagram

Fig. 3. The back-end of the SSRS

The radar back-end provides the digital signal processing unit, the radar front-end, the camera and the rotary unit with the necessary power. The digital signal processing unit is based on the PC/104 standard and consists of two analog to digital converters (ADC), a digital I/O hardware (DIO) and a computer (CPU). The ADC digitizes the analog intermediate signals coming from the radar front-end. The DIO is used for timing - and radar front-end control. Up to four different chirp parameters can be chosen via the DIO. Additionally, the power of the radar front-end can be controlled digitally. The motor controller drives the rotary unit and triggers the chirp sequence in every rotation. The CPU handles the data of the ADC and DIO, communicates with the rotary unit controller and is used as an interface between the system computer and the radar back-end. Table II shows technical specifications of the radar back-end. The maximum measurement range depends on the chosen bandwidth of the radar, since the number of samples is fixed. The example shown in Table II shows the chosen value for the measurement campaign. With the maximum bandwidth of 1 GHz shown in Table I, the maximum range is thus reduced accordingly.

TABLE II
Specifications of the radar back-end

Maximum measurement range	e.g. 256 m at 300 MHz bandwidth
Maximum image refresh rate	1.6 images per second
Sensor configuration	4 different chirps
Physical dimensions	345 × 275 × 320 mm^3
Weight	12904.5 g

3. System computer

The system computer is a standard notebook and is connected to the radar back-end and to the camera via 1 GBit/s Ethernet to transfer the raw data of the system. Additionally, the system can be controlled with the system computer's graphical user interface. To create range-Doppler-maps of the radar data, the system computer has to perform two-dimensional fast Fourier transformations. Moreover, the monopulse processing for 3 dimensional localization (3D), the tracking of up to 4 targets on ground or in the air, the Doppler analysis for classification, the calibration and RCS analysis is done with this computer. Furthermore, the PC visualizes the processed radar data - and the camera images on the screen. Figure 4 shows the data plot of the graphical user interface. In this example, two targets (drones) are detected and tracked by the software.

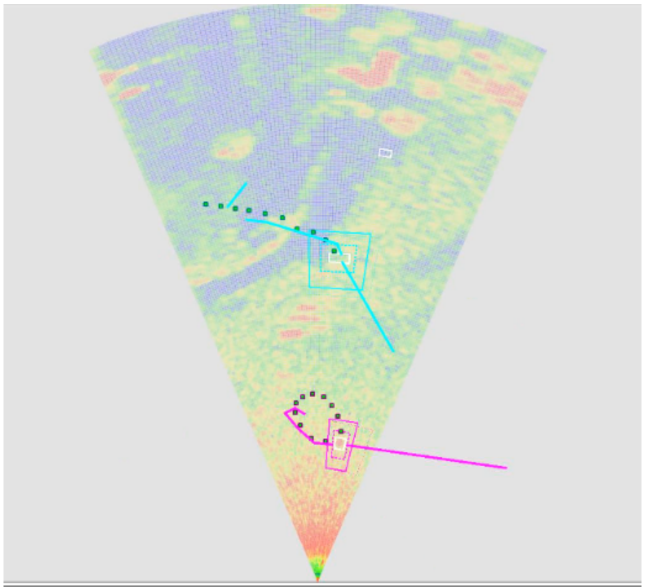


Fig. 4. Data visualization of the SSRS. The ring segment shows the observed area with the tracked targets.

B. Measurement Campaign

This section describes the measurement setup consisting of the correct configuration of the radar system, the description of the measurement area and a list of the measured objects.

Before doing the measurements, the radar system must be configured correctly. Table III shows the parameter set used for the measurements, whereby the parameters are optimized for range Doppler processing in real-time.

TABLE III
Parameter set of the SSRS for the measurement setup

Radar frequency	94.252 GHz
Radar bandwidth	300 MHz
Range resolution	0.5 m
Measurement update rate	1.6 Hz
Velocity resolution	0.7 m/s
Chirp repetition frequency	18.473 kHz
Maximum target speed	±52.92 km/h
Samples per Chirp	1024
Maximum range	256 m
Sample frequency	19.32 MHz
IF-Bandwidth	9 MHz
Antenna Beamwidth	1.3° x 13°
3D surveillance volume	256 m ×46° × 18.5°

The radar bandwidth of 300 MHz results in a range resolution of 0.5 m. The velocity resolution of 0.7 m/s depends on the measurement update rate of 1.6 Hz and the antennae. The chirp repetition frequency of 18.473 kHz leads to the maximum radial target speed of ±52.92 km/h. The maximum range for this configuration is 256 m and the resulting maximum intermediate frequency is 9 MHz. Therefore, the required sampling frequency of the analog to digital converter is 19.32 MHz. The 3D surveillance volume is 256 m in range, 46° in azimuth and 18.5° in elevation.

The measurements were conducted in the autumn of 2020 on a measurement area of the Fraunhofer FHR in Wachtberg. Figure 5 shows a photograph of the measurement area.

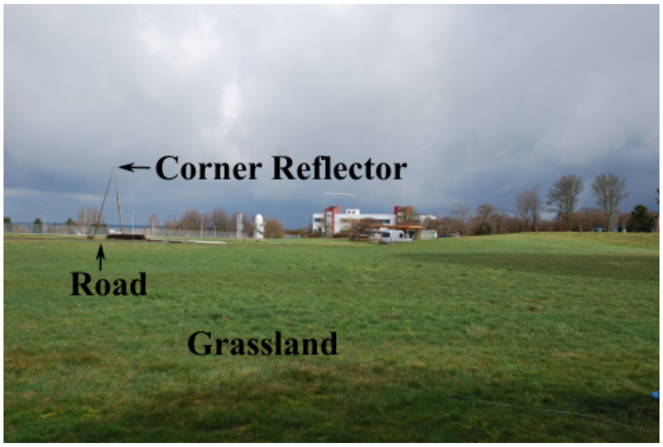
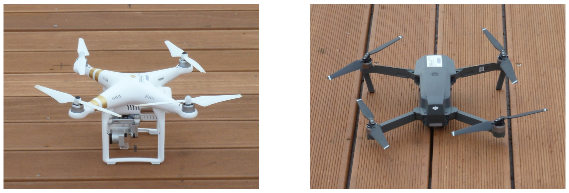


Fig. 5. Measurement area used for the measurements of drones with the SSRS

The measurement area consists mainly of grassland with a corner reflector in the back and a narrow road.

For the evaluation of the data, three data sets of the full measurement that was conducted at this day are used. These data sets are selected due to the exact knowledge of the target positions, which are used later as ground truth data. Two of these three scenarios actually show two persons standing on the grassland. In one measurement, one person is holding a corner reflector with an RCS of $10\,\mathrm{m}^2$. The remaining scenario consists of two drones hovering at the same position one above the other. The two drones are a DJI Phantom and a DJI Mavic, which both are categorized as micro drones [1]. These micro drones are typically small with a weight of less than $2\,\mathrm{kg}$ and possess a small Radar Cross Section (RCS), which makes the detection more demanding [5], especially close to the ground that produces a large echo itself. The weight of the Mavic is actually below $1\,\mathrm{kg}$, i.e. $895\,\mathrm{g}$ with a diagonal length of $380\,\mathrm{mm}$. The Phantom possesses a weight of $1216\,\mathrm{g}$ and a diagonal of $350\,\mathrm{mm}$ without rotor[1]. Both drones are depicted in Figure 6. An overview of the data that is used in this work is given in the following section.



(a) DJI Phantom III

(b) DJI Mavic

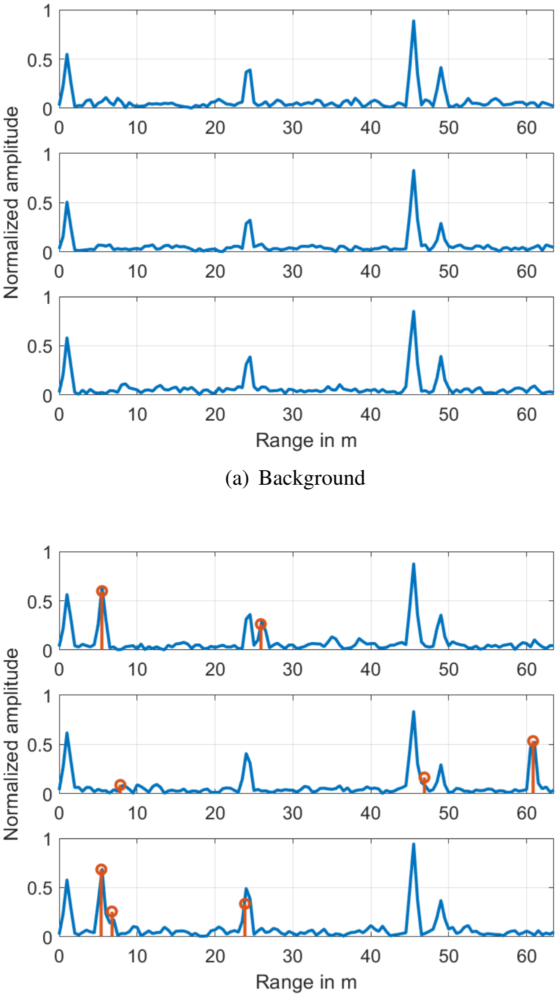
Fig. 6. Drones used in the measurements

III. Data Description

As it was already mentioned in the introduction, measurements of drones and persons are available to validate the algorithm. Additionally, a simulation is used to create a scenario with known background and targets with a larger dynamic variation than in the measured data.

A. Simulated Data

The first data set consists of simulated High Range Resolution (HRR) profiles, which consists of a total of $5\,000$ background profiles and $50\,000$ profiles with targets. The background profiles always contain four point targets at fixed positions, each of which with a variation of $\pm 10\%$ in amplitude. These four scattering centers represent fixed targets in the background that can cover small targets. The nominal amplitudes of these scattering centers are 0.5603, 0.4632, 0.8954, and 0.3857 respectively. The variation in each of the scattering centers is similar to the variation that is observed in the measured data described later. For the larger part of the data, one to a maximum of three targets with random amplitude at random position are added to this background. The amplitudes of the scattering centers in the test data is equally distributed between 0.1 and 1. Examples of both types of profiles are shown in Figure 7, where for the profiles with targets, these are included as red circles at the corresponding position. The input data of the network are the profiles drawn in blue. The radar parameters of this simulation are set to the values given in Table III, i.e. a carrier frequency of 94.252 GHz and a bandwidth of $300\,\mathrm{MHz}$. However, since this is a solely signal processing based simulation and phase information is not used here, these profiles can also be created with any other setting of parameters. The range scaling would change with other parameters, but the input of the autoencoder, i.e. the profile with its amplitude values would be the same. The simulated data is created in MATLAB using a basic stretch processing simulation with point targets similar to the simulation provided by [15].



(a) Background

(b) Background with targets

Fig. 7. Examples of the simulated HRR profiles. Plot (a) shows the simulated background, which consists of four scattering centers plus noise. Plot (b) shows this background with targets at random position and with random amplitude labeled by a red circle.

In the simulated background profiles in Fig. 7 (a), four constant targets with slightly varying amplitude around the nominal value mentioned above can be seen. In the profiles with targets in Figure 7 (b), three typical scenarios are depicted. In the top profile, two isolated and relatively strong scattering centers can be seen. In

[1] The information about the weight and size of the drones is from the website of the manufacturer.

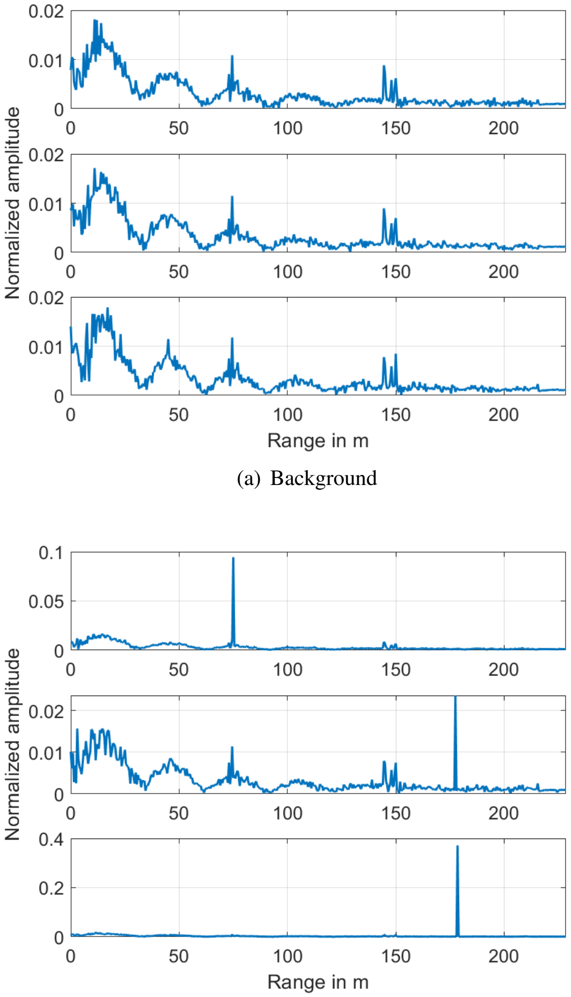
the middle profile, two very weak targets and one strong target are present. The second weak target around $48\,\mathrm{m}$ is additionally covered by the strong background target next to it. These types of targets certainly represent the most difficult targets to detect. In the third profile, a target coincides with a background scattering center, increasing the amplitude at this point. The other two scattering centers in the third profile appear close to each other as a single target spread over several range cells.

From the $50\,000$ profiles with targets, $45\,000$ were used in addition to the background profiles to train the network and $5\,000$ were used to test the algorithm.

B. Measured Data

Several scenarios with drones and people were measured, as it was described in section II. To validate the method, two measurements with people and one with drones are used, since the targets did not move during these measurements and thus exact knowledge about their location is available. To calculate the average background, two measurement runs are used in which only the background without targets was measured, resulting in 395 range profiles of a specific angle without targets. To increase this amount of data, peaks with a random amplitude and a random position are included in the background profiles. These peaks represent ideal point targets that do not move, i.e. no Doppler information is added to the target. As mentioned above, the targets in the chosen data also do not move and thus produce no usable Doppler information. The rotation of the blades of the drones is not considered here. The choice of only peaks and no other targets in the training data is based on the purpose of the algorithm, i.e. the detection of small targets.

Examples of background measurements and background with artificial targets can be seen in Figure 8. The magnitude of the range profiles is normalized to the maximum value that appeared during the complete measurement campaign, therefore the magnitude of the chosen range profiles with the rather small targets are comparably low. The maximum amplitude of an artificially added point target is 0.4, which is comparable to the last profile in Figure 8 (b). The smallest value of the added point targets is 0.01, which is comparable to the surrounding clutter visible in the background. The amplitude values are chosen to create targets in the expected dynamic range of the measurements. This should allow a training of the autoencoder to properly learn the reconstruction of the scene with targets. In total, the amount of data is increased by a factor of 20 with randomly positioned peaks in the background data. Accordingly, a total of 20 new profiles are generated from each profile with randomly selected targets. Finally, a total of $8\,295$ training profiles are available.



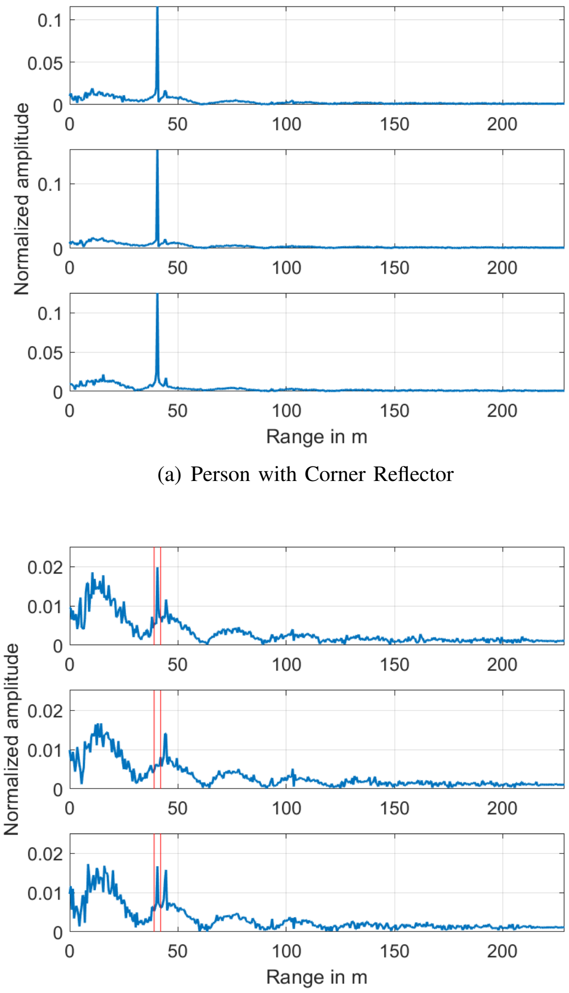
(a) Background

(b) Background with artificial point targets

Fig. 8. Examples of the measured HRR background profiles

As already mentioned, two measurements with persons are used for evaluation, since in this case the exact position of the target is known and thus a detection and false alarm rate can be calculated. In the first measurement, the person is holding a corner, which is the reason why the person is very clearly visible in the profiles shown in Figure 9 (a). This measurement is only used to evaluate if the method works in principle and if the network has learned the scene correctly. It can be considered as a toy example in this context. The second measurement was made without the corner reflector and corresponds therefore to the real measurement situation. In the latter case, the person is already very difficult to be recognized by a human observer in the measured profiles sometimes. Examples of both measurements are shown in Figure 9, where in the measurements without corner reflector the area of the target is marked with red lines.

The profiles without corner reflector in Figure 9 (b) show how weak the echo of the person is in the measured profile and how large the variation is. In the upper profile, the echo is still quite strong, while in the middle profile it is practically invisible. In the lower profile, the echo is comparable to an echo of the background immediately behind the target. The measurements with corner show a



(a) Person with Corner Reflector

(b) Person without Corner Reflector

Fig. 9. Examples of the measured HRR profiles with person. The person, i.e. the target that has to be detected, is between the two red lines in plot (b). In plot (a), the target is at the same position, but clearly visible due to the strong echo of the corner reflector.

sharp increase at the position of the target, which makes the target clearly visible. The target was a male person that was supposed to not move during the measurement and stand as still as possible.

A second example with nonmoving targets of the measurements described in section II is the scenario with the two drones hovering at the same position. In this case, it is also possible to calculate false alarm and detection rates, since the targets do not move and thus ground truth information is available. Some examples of the measured HRR profiles are given in Figure 10. The range cells of the two drones are again marked by two red lines to show the variability of these two targets and the dynamic of the target is comparable to the scenario with the person. Since Doppler is not considered in this investigation, the rotation of the blades is not exploited for detection. The two drones are depicted in Figure 6.

It should although be mentioned again that no integration is used to emphasize the targets in the profiles. This would be possible since the targets did not move, but the absence of any motion is only to generate ground truth information about the position without any additional tracking. In general, the algorithm is able to detect moving targets, as long as the background does not change significantly.

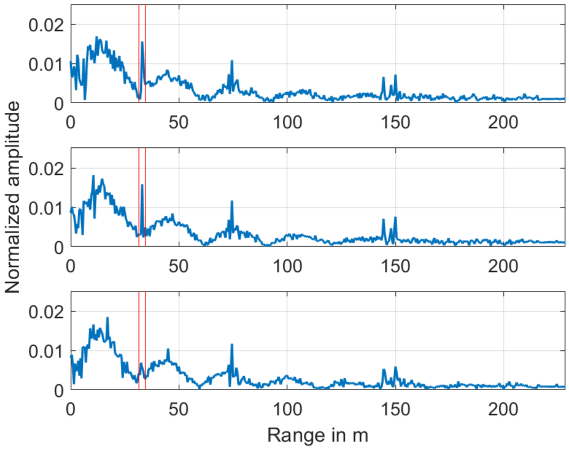


Fig. 10. Examples of the measured HRR profiles of the scenario with two hovering drones. The targets are in the area between the two red lines.

IV. Proposed Detection Scheme

The detection scheme presented in this paper is based on a neural network called autoencoder, which is described in section A. The detection algorithms are described in sections B and C.

A. Autoencoder

One property of machine learning methods is automatic feature extraction in measured data, which eliminates the need for time consuming manual analysis and selection of features. This property is also called representation learning [16] and an autoencoder uses it to create a code as internal representation of the input data. The output of a basic autoencoder should be the same as the input value, i.e. a reconstruction based on the internal representation. The dimension of this code defines the basic structure of the autoencoder. In case of a dimension smaller than the dimension of the input signal, the autoencoder is called *undercomplete*. In this case, an intrinsic data compression is applied, which results very likely in an imperfect reconstruction. The basic structure can be seen in Figure 11.

In this work, undercomplete autoencoders are used, since on the one hand the required computation time is reduced, and on the other hand the mentioned data compression should increase the robustness against noise. The actual size of the used autoencoder will be given below.

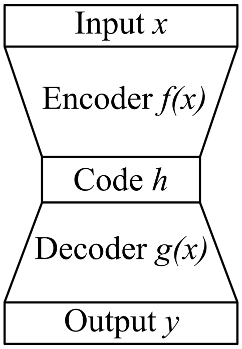


Fig. 11. General structure of an undercomplete Autoencoder.

Figure 11 also shows that an autoencoder essentially consists of two elements, an encoder and a decoder. The encoder generates the code from a given input signal, while the decoder tries to reconstruct the input signal from the code. Both elements can be formally represented as

$$f(x) = h = \varphi_f(W_f x + b_f) \tag{1}$$

$$g(h) = y = \varphi_g(W_g h + b_g), \tag{2}$$

where the variables W_f and W_g represent the weights and b_f and b_g the bias of the neural network. These are the free parameters of the network, which have to be determined during training.

The functions $\varphi_f(\cdot)$ and $\varphi_g(\cdot)$ represent the activation functions of the neural network and determine the range value of the code and of the reconstruction. In the networks used here, the leaky Rectified Linear Unit [17] (leaky ReLU) is used to avoid a stop of the backpropagation at neurons with a negative output, which would result in *dead neurons*.

Further details on these activation function, the used training algorithm called *Adam*, and on the general structure of neural networks can be found in textbooks such as [9].

Two detection schemes are proposed for the datasets, which are introduced in the following subsections.

B. Detection Scheme for the Simulated Data

In this work, we exploit a neural network that is inspired by [18]. It is a 1-dimensional fully-convolutional autoencoder, with a number of sixteen layers, where the encoding and decoding part are symmetric to each other. On the one side, the encoder is comprised of three blocks, where each block contains two 5x1 convolutional layers followed by LeakyReLU activation functions. The kernels are chosen of size five so that sufficient amount of correlation between adjacent range bins in the profiles is captured by the network. For comparison purposes, kernel sizes three and seven have been tested and concluded that deterioration in the detection results is noted for the former kernel size, however there was no difference obtained in the latter case, when compared to size five. A larger kernel size than the one selected here would be necessary in cases such that bandwidth of the radar waveform is increased, given that the range resolution reduces. The spatial dimensionality inside the blocks is preserved, i.e. there is no present stride component. Nevertheless, the downsampling step is performed in between the blocks by the use of strided convolution. On the other side, the upsampling in the decoder is achieved by the use of transposed convolution. One important part of the network are the skip connections; the high resolution feature maps of the encoder are fetched to the decoder to allow fine grained details being recovered in the detection. The chosen architecture achieves good generalization to unseen data and, therefore, avoids the overfitting issue.

A scheme for the detection network is given in Figure 12, where C is the cost function defined as the mean squared error (MSE) between the reconstruction y and the target amplitude A. The network is trained for 100 epochs and the number of samples fed to the network, i.e. the batch size is set to 32.

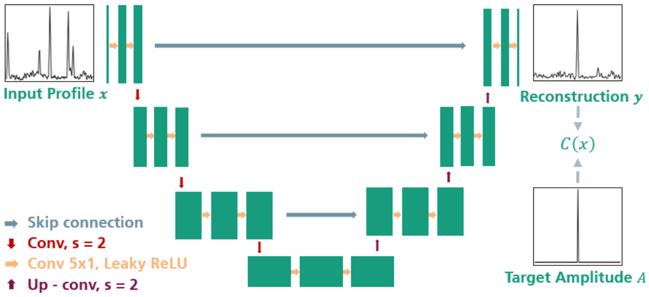


Fig. 12. Detection scheme of the simulated data.

The actual detection takes place in the reconstructed profiles with a threshold value. Unfortunately, the automatic determination of the threshold value could not yet be solved satisfactorily, which is why the detection and false alarm rates are presented here using a variation of the threshold value and the common Receiver Operating Characteristic (ROC) curves. The detection algorithm performed in the case of simulated data can be described as following:

1) A set of labels that contains the amplitude of the targets in the data is created.
2) The autoencoder is being fed with the training set of profiles to learn adequate parameters.
3) The loss function that is used, MSE calculates the squared distance between the reconstructions and the new set of labels.
4) An unknown test profile is fed into the trained autoencoder.
5) Threshold detection is performed in the reconstruction.

C. Detection Scheme for the Measured Data

Due to the lack of sufficient amount of training samples in this dataset, our convolutional autoencoder does not generate the desired results in detection. Therefore, a different approach is chosen for the measured dataset. The autoencoder used in this case is fully connected and has a much lower number of layers, namely two. The code layer possesses 343 neurons, which is about 75% of the input size. This compression ratio was determined experimentally and gave a good compromise of reconstruction quality and noise robustness. However, an extensive parameter grid search, also for the other parameter mentioned later on, might give a different structure with a better performance.

The chosen activation functions are respectively leaky ReLU at the encoder and a linear at the decoder side. The linear activation function in the decoder allows a reconstruction of arbitrary input values, but since the used data is magnitude data, a rectified function like the leaky ReLU would also be possible. However, the initialization with random weights for the training might give some dead neurons in this case at the output of the network. That would take much more iterations than with the linear activation function to change these neurons to a positive output value. Therefore, the linear activation function is preferred in the output layer during the training. Nevertheless, in the application, the linear activation is replaced by a ReLU function to remove unwanted negative values in the output signal. The reason for this will be given below.

The cost function of the basic autoencoder is presented in the following parts.

1. *Contractive Autoencoder*

The used variation is called *contractive autoencoder* and, as it was already mentioned in section I, is designed to give a code that is invariant to small changes in the input data [19]. Therefore, it should be robust against noise and the number of false alarms should be decreased. The principle of this autoencoder is shortly described by the used cost function

$$C(x) = \frac{1}{2}||y - x||_2^2 + \lambda||J_{f(x)}||_F^2, \tag{3}$$

which consists of two terms. The first term represents the quality of the reconstruction, which is always part of an autoencoder cost function. Either in this form, i.e. the squared error, or in a different one, e.g. a cross entropy term. The second term represents the squared Frobenius norm of the encoder's Jacobian matrix, i.e. the sum of the squared partial derivatives of the code layer's neurons with respect to the input data

$$||J_{f(x)}||_F^2 = \sum_{i,j}\left(\frac{\partial h_j(x)}{\partial x_i}\right)^2. \tag{4}$$

Thus, a high value of this term represents a large change in the code with a change in the input data. The calculation of the Jacobian matrix and its derivative is of the same complexity as the standard backpropagation [19] and a part of the backpropagation can actually be reused in the calculation of the Jacobian and its gradient.

The determination of the weight λ of the contractive term in the cost function of network is a crucial step in design of the network. A large weight would suppress clutter with high variations, but would also suppress small targets. A weight that is too small would create many false alarms, since small variations in the clutter could trigger detections. In this work its determination is done in an adaptive way that keeps the ratio of the two terms constant. Thus, the two cost terms $\frac{1}{2}||y - x||_2^2$ and $||J_{f(x)}||_F^2$ are first determined and then λ is chosen so that the first and second summation terms of the cost function are in a fixed ratio. This allows the training to focus on the reconstruction and weigh the contracting property as specified. Moreover, the ratio of these two terms is increased during the training phase in order to control the priorities of the network. This allows to focus on the reconstruction at the beginning of the training and increase the relevance of the contractive term towards the end. In our implementation, the ratio started with a value of 10^{-4} and is increased by 3% at the end of each epoch. The training time was set to 100 epochs. These values were chosen experimentally.

In [19] it was also shown that the contraction happens mainly around the given training examples and that areas away from the training samples in input space possess less contraction. This is necessary to achieve a useful feature extraction instead of a global scaling effect. This behavior confirms the intention of a robust feature extraction, but shows also the need for representative training data. This geometric interpretation of a contractive autoencoder can be used to describe the algorithm in the way that the contraction happens around the clutter in the scene, since this is already present in the background data. The new targets should be outside this contraction area and should be present in the reconstruction. The artificial targets in the training data create also a contraction outside the background data, but are necessary to allow a proper reconstruction in the expected dynamic range. To emphasize the targets again in the reconstruction, a retraining described below is performed without the contractive term. Further information on the geometric interpretation of this kind of autoencoder can be found in [19].

2. *Bias Freezing*

A deviation from the commonly used training methods is the use of the bias in the decoder of the network. Usually, the bias is a free parameter and is learned during training. However, since the system used here is designed for a given background, the background averaged over time is used as bias and cannot be changed during training. After training, the bias is removed from the network and only deviations from the mean background should be reconstructed. This is similar to the idea of background subtraction, but the input data is processed by the autoencoder, which should have learned the statistical properties of it. Furthermore, the contractive autoencoder should reduce the variations in output due to the behavior described above. In case of background subtraction, the variation in the output is simply the difference to the mean value.

To the knowledge of the authors, this is the first time that this freezing method is used for a detection algorithm. Other publication freeze complete layers or nodes to accelerate the training [20] and later remove nodes without significant influence on the output [21].

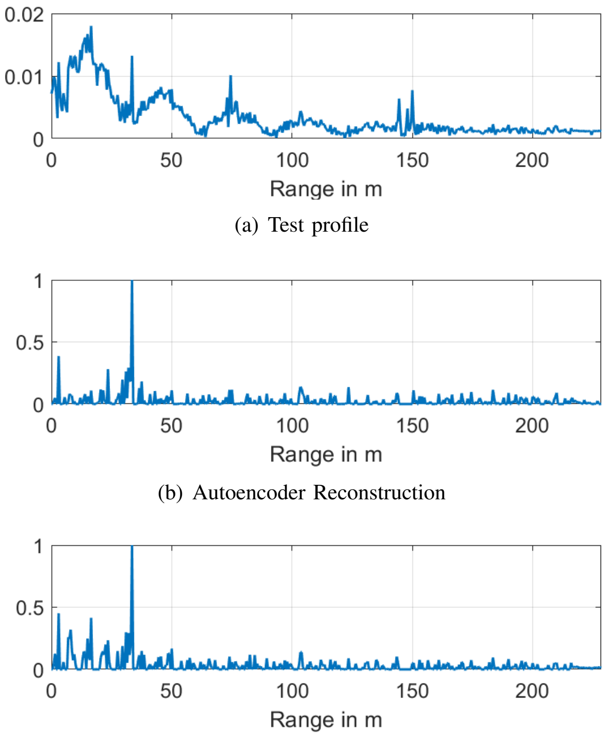
3. *Retraining of the Autoencoder*

To further improve the denoising behavior of the network, another training step is performed after the bias has been removed. For this second training, the input data is again the background data with artificial targets as it was shown in Figure 8 (b). Opposite to the first training, the target output of the network is not the input data anymore, but a vector with only one nonzero element at the position of the included target that contains the amplitude of that scattering center. After this ”target only” retraining, the network shows an improved clutter suppression. A comparison of the detection performance with and without this retraining will be given in Section V.B.

For this retraining, the learning rate is reduced from 10^{-3}, which was used in the first training to 10^{-4} and the contractive term in the cost function is removed. The reduced learning rate is used to create only small changes in the network, since the actual reconstruction has already been learned in the first training. The contractive term has been removed, since an analysis of the output after the first training has shown the desired clutter reduction, but also a reduced amplitude of unknown targets in the test data. Therefore this term has been removed to increase the dynamic in the data again.

It was already mentioned above that the activation function of the output layer is changed to a ReLU function for testing. The reason for this change are the negative output values, which are produced after the bias term is removed. These negative values are not relevant for a detection and are thus discarded. For a better comparison between the created profiles, the negative output values of the reference method, i.e. the background subtraction, are also set to zero.

As an example, the reconstruction of a later used range profile is depicted in Figure 13. This Figure shows the original profile with the corresponding profiles created by mean background subtraction and the reconstruction by the autoencoder. The profile contains a single target, which is clearly visible in both detection profiles. The profiles in Figure 13 (b) and (c) have been normalized to the interval [0, 1] for a better comparison between them. In the calculation of the actual result, the profiles are not normalized. The amplitude of the original profile in Figure 13 (a) is still normalized to the maximum value that appeared in the measurement campaign. The difference between the autoencoder reconstruction and the background subtraction can be seen in the areas without targets, which shows for the autoencoder output an almost constant noise, while the noise in the background subtraction decreases with increasing range. Therefore, the mean background subtraction method might create more false alarms close to the radar and the autoencoder at the end of the observed range. Nevertheless, since the magnitude of the background subtraction method is significantly higher in areas without target than for the autoencoder method, the former is more likely to create false alarms. The influence of the Gaussian background subtraction will be shown in Section V.B together with an analysis of the target to background dynamic for the measured data.



(a) Test profile

(b) Autoencoder Reconstruction

(c) Background Subtraction

Fig. 13. Measured profile with autoencoder reconstruction and mean background subtraction

Before the results are presented, the algorithm for the measured data is summarized briefly:

1) An averaged background is determined from the independently measured background data.
2) The decoder bias is initialized with the averaged background and cannot be changed during the training.
3) The autoencoder is trained with the available training data, i.e. the free parameters of the network are determined.
4) The bias of the decoder is set to zero.
5) A second training with a reduced learning rate and no contractive term is performed with a desired target only output.
6) An unknown test profile is fed into the trained autoencoder.

7) Threshold detection is performed in the reconstruction.

The autoencoder is applied to a single profile of the ring segment, which is measured by the radar and shown in Figure 4. Therefore, if this algorithm is used for the full scene, each azimuth angle requires the training of an independent autoencoder. In principle, the training of a single autoencoder for a two-dimensional input is also possible, but since the amount of available training data is very limited and the size of such a network would be very large, the proposed solution is preferred. A graphical representation of the algorithm is depicted in Figure 14, where the decoder function is shown without activation function $\varphi(\cdot)$ during training, since a linear activation function is used. For testing, a ReLU activation function is used to remove unwanted negative values.

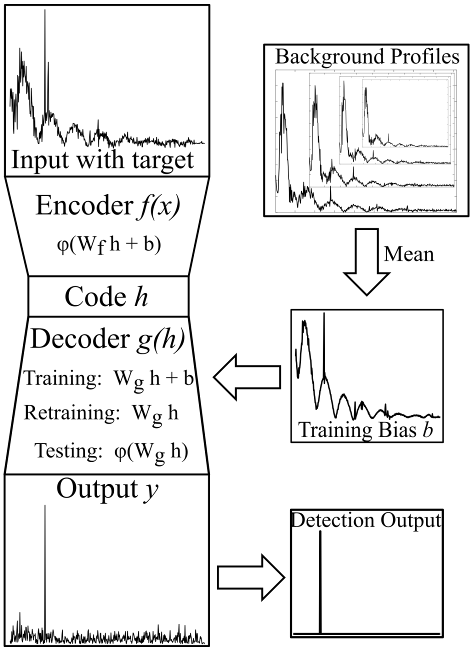


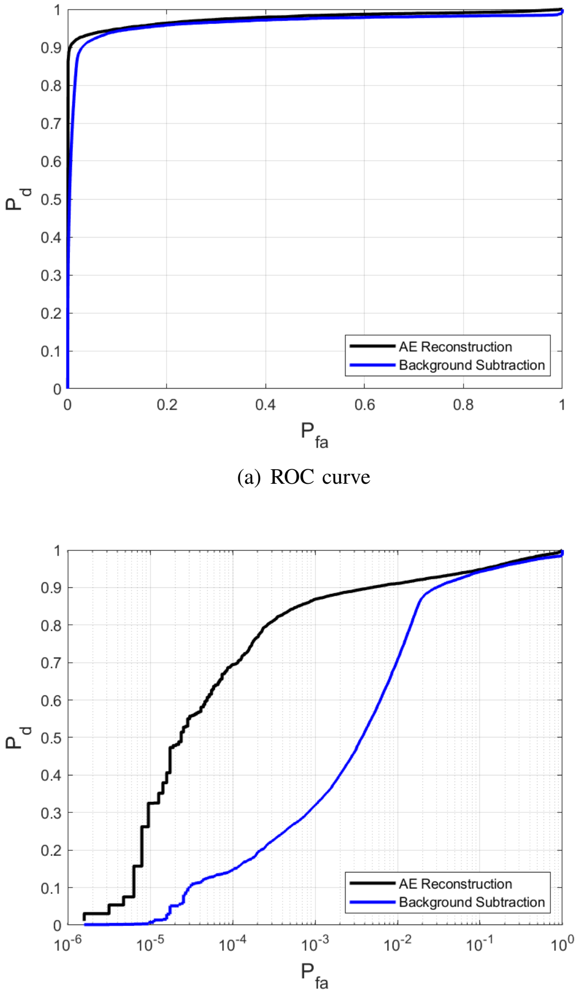
Fig. 14. Graphical flowchart of the algorithm. The mean value of the measured background profiles are used during training in the decoder and is removed for testing. The output shows a reconstruction of a test profile without the bias and in the lower right corner, the actual detection result is depicted.

V. Results

In this section, we present the results obtained with our data. The results will be mostly presented as ROC curves for the different data sets. These curves are calculated in the way that the number of range cells multiplied with the number of range profiles gives the number of possible detections. The number of correctly detected targets divided by the number of possible detections gives the detection rate P_d. The number of detections outside the target positions divided by the number of possible detections gives the false alarm rate P_{fa} and the combination of P_d and P_{fa} defines one point in the ROC curve. The detection itself is a basic fixed threshold detection over the full range profile and is performed for 2.000 thresholds. The signal to noise ratio is not varied for the simulated data and also not estimated for the measured data. It is taken as given here.

A. Simulated Data

The ROC curve for the entire data set is shown in Figure 15, i.e. the detection and false alarm rate averaged over all targets of the test data set. The ROC curve of the mean background subtraction is shown in blue and the curve of the autoencoder is printed in black.



(a) ROC curve

(b) Logarithmic ROC curve

Fig. 15. ROC curves of the simulated data

The proposed method with the autoencoder performs better than the reference for low false alarm rates. As it can be seen in Figure 15 (a), the black curve is above the blue curve of the background subtraction and closer to the vertical axis, which means that there are fewer false alarms created for a certain detection probability. To take a closer look at this case, we show the logarithmic ROC curve in Figure 15 (b). The autoencoder has already

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3253469

generated some detections before the first false alarms appear, and it continously achieves a higher detection rate for low false alarm rates than the background subtraction. The superiority of the autoencoder is clearly visible for $P_{fa} < 0.02$. For high false alarm rates, i.e. $P_{fa} > 0.02$, the methods are comparable. However, the black curve of the autoencoder is slightly higher and the targets are detected earlier than in the reference method. These results are supported by the values of the area under the curve (AUC), which are 0.9764 for the autoencoder and 0.9632 for the background subtraction.

B. Measured Data

To evaluate the results of the measured data, two different measures are used. The first one is a comparison of the peak signal to clutter ratio (SCR) in the output profiles. To calculate this ratio, the highest value of the target area is divided by the maximum of the area without target

$$SCR = \frac{\max\left(y_{\text{target area}}\right)}{\max\left(y_{\text{no target area}}\right)}. \tag{5}$$

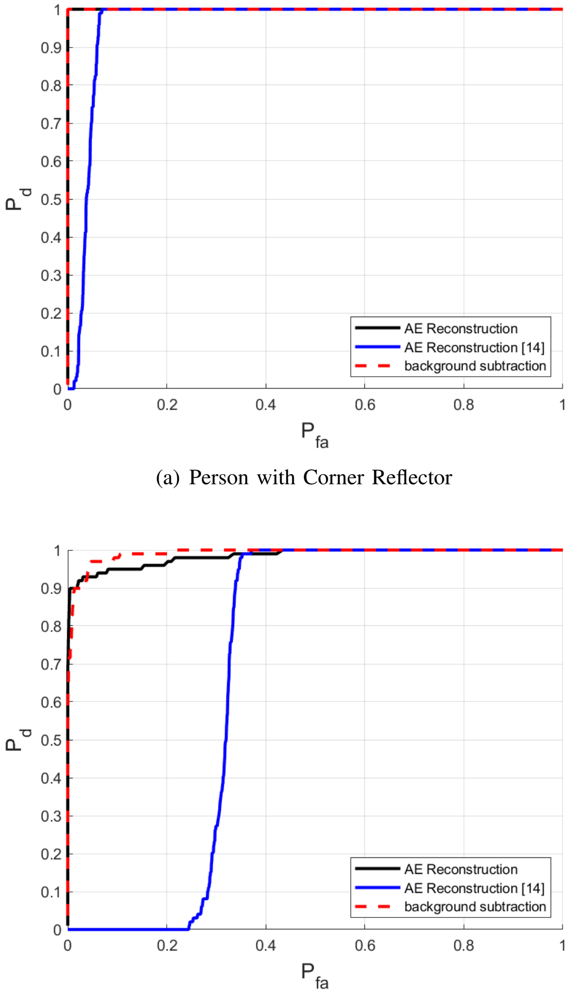
This ratio is calculated here for the mean background subtraction and the autoencoder reconstruction and afterwards a comparison is done to determine the gain of the proposed method. In Table IV a summary of the results is given for mean background subtraction (BS) and the autoencoder (AE). For each data set, i.e. corner, person and drones, the following parameter are given: the minimum SCR SCR^{BS}_{min} and SCR^{AE}_{min}, the maximum SCR SCR^{BS}_{max} and SCR^{AE}_{max}, the mean SCR $\overline{SCR}^{BS}$ and $\overline{SCR}^{AE}$, the number of profiles with the maximum outside the target area $\#\{y^{BS}_{fa}\}$ and $\#\{y^{AE}_{fa}\}$, and the average gain of the autoencoder reconstruction compared to the background subtraction $\overline{SCR}^{AE}_{gain} = \overline{SCR}^{AE}/\overline{SCR}^{BS}$.

TABLE IV
Comparison of the SCR for backgound subtraction (BS) and autoencoder reconstruction (AE)

	Corner	Person	Drones
SCR^{BS}_{min}	7.02	0.10	0.05
SCR^{AE}_{min}	12.30	0.03	0
SCR^{BS}_{max}	41.11	6.36	8.73
SCR^{AE}_{max}	36.26	11.67	9.60
$\overline{SCR}^{BS}$	18.64	1.83	1.84
$\overline{SCR}^{AE}$	24.30	3.19	2.85
$\#\{y^{BS}_{fa}\}$	0	35	30
$\#\{y^{AE}_{fa}\}$	0	21	17
$\overline{SCR}^{AE}_{gain}$	1.39	1.67	1.71

The results show that the autoencoder reconstruction possesses a higher target to background dynamic in average, although the minimum SCR is close or exactly zero in the two relevant data sets. However, in case of the challenging targets, i.e. the drones and the person, the maximum and the mean SCR is clearly increased with a gain of 67% and 71% respectively. Furthermore, the number of profiles with the maximum value outside the target area is clearly reduced from 35 to 21 and 30 to 17 cases. These numbers can be explained with the behavior of the network that was shown in the example in Figure 13.

To evaluate the influence of the noise floor that is visible in the autoencoder reconstruction, ROC curves for the different data sets are calculated for a detection with a fixed threshold. In Figure 16 the ROC curves for the corner and person data set are shown. The ROC curve of the background subtraction is shown as dashed red line and the curve of the autoencoder is printed in black. As it was mentioned in Section I, the detection results using the algorithm from [14] applied on this scene is also shown here in the blue curve.



(a) Person with Corner Reflector

(b) Person without Corner Reflector

Fig. 16. ROC curves of the measured profiles with persons

As it can be seen from the ROC curves, the detection method works perfectly in the case of the corner reflector for the algorithm presented here and the mean background subtraction. The target is detected in all measurements without triggering any false alarms. The algorithm from [14], which was presented as a proof of concept to show that a detection with this kind of algorithm is possible in principle, has already some problems with the corner data and is falling far behind in the data without the corner. The latter is also the more interesting case and it can be seen that the black curve of the autoencoder is below the dashed red curve of the background subtraction. For a closer look at the areas of small false alarm rates, the ROC curves for the person without corner are plotted with a logarithmic x-axis in Figure 17. The blue curve of the former method is not shown anymore since it is not competitive and the red curve is now plotted as a solid line, since the overlap with the black curve is reduced.

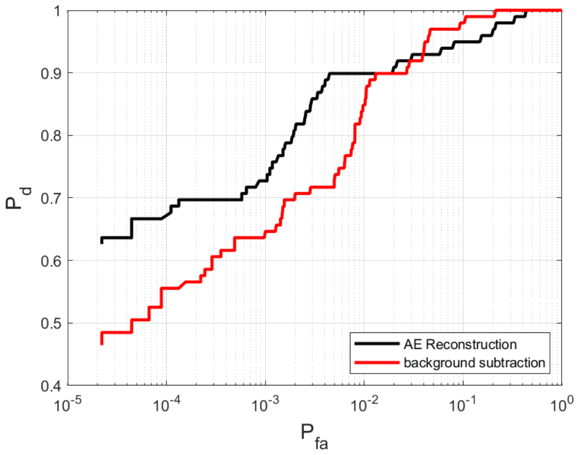
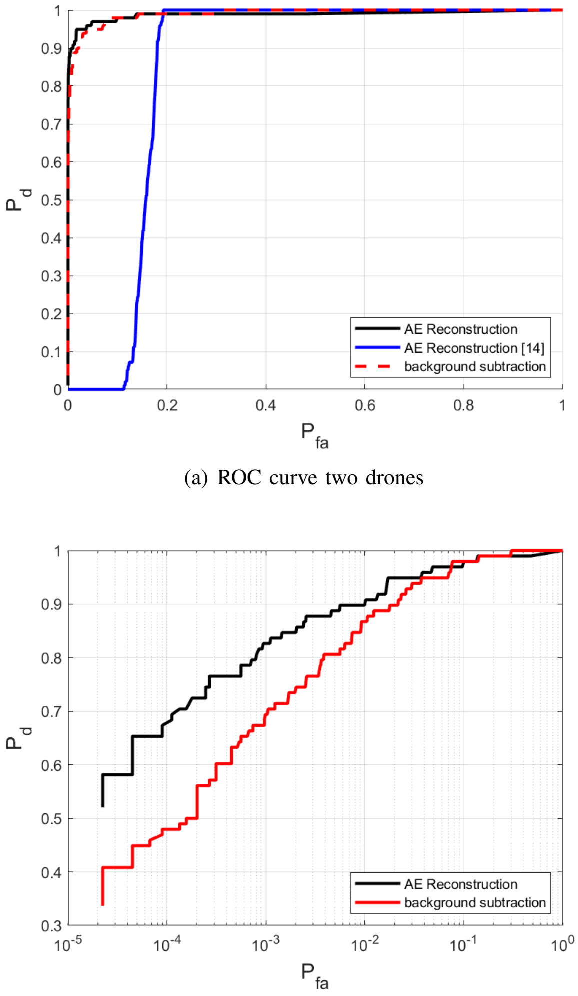


Fig. 17. Logarithmic ROC curves of the measured profiles with persons. The plots begin at the false alarm rate that corresponds to a single false alarm in the test data.

The curves show that for the autoencoder, the first false alarm appears after the target has been detected in more than 60% of the profiles. In case of background subtraction, the detection rate is below 50% when the first false alarms appear. The numerical value around $2 \cdot 10^{-5}$ corresponds to a single false alarm for the given number of range cells in all test range profiles. Up to a detection rate of 90% and low false alarm rates, the autoencoder is above the background subtraction. Around a false alarm rate of 10^{-2}, both methods are comparable and for high false alarm rates, the background subtraction is clearly better. This is also confirmed by the AUC, which is 0.9915 for the background subtraction and 0.9841 for the autoencoder. The high number of false alarms in the autoencoder reconstruction are created by the higher noise floor in the recontruction profiles shown in Figure 13. For low thresholds, this noise floor causes a large number of false alarms compared to the background subtraction method.

Next, the results obtained for the drones test case are shown. In this case, the linear ROC curve in Figure 18 (a) shows a comparable behavior of the proposed method and the mean background subtraction, although the 100% detection rate is achieved earlier with the mean background subtraction. The algorithm from [14] is again not competitive. For better visualization, the ROC curve is shown in Figure 18 (b) in logarithmic scale similar to the example above. The absolute detection rate is lower compared to the example above, but the autoencoder reconstruction gives around 20% more detections when the first false alarms appear and is above the background subtraction curve up to a false alarm rate of almost 0.1. The AUC is comparable to the example above, 0.9906 for the mean background subtraction and 0.9882 for the autoencoder method.



(a) ROC curve two drones

(b) Logarithmic ROC curve two drones

Fig. 18. The resulting ROC curve of the scenario with two hovering drones

It was already mentioned above that Gaussian background subtraction is used as additional reference method. The implementation here is an adapted version of [22] that calculates a window $g(r)$ for each range profile dependent on the standard deviation of the known background data. To calculate the window, the profile created by the mean background subtraction is divided by the standard deviation of the background data. Afterwards, two thresholds are used to identify a range cell as foreground or background. If the ratio at range cell r_i is larger than the foreground threshold M, the window gets the value 1, i.e. $g(r_i) = 1$. If the ratio is below the background threshold m, the range cell is considered as background and $g(r_i) = 0$. If the ratio is between these two thresholds, an intermediate value is calculated. For further details, the reader is referred to [22].

For the results here, the thresholds are chosen as given in the original paper as $m = 1$ and $M = 2.5$. After the window is calculated, it is multiplied with the profile created by the mean background subtraction for a stronger suppression of the areas with large variations in the background. The original algorithm uses a moving window to calculate the mean background to adapt to a changing background. Since the background in the scenario chosen here is fixed, but shows some fluctuations, the standard deviation is calculated only once with the independently measured background data.

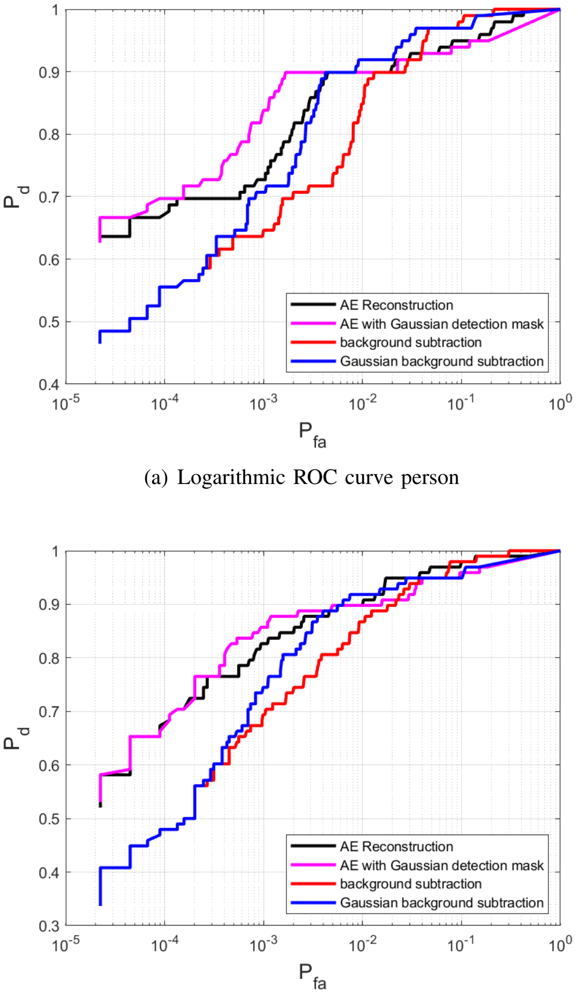
This kind of window function can also be applied to the autoencoder reconstruction to suppress the noise in the autoencoder output. Therefore, the standard deviation in the reconstructions of the background data is calculated and a window is calculated in the same way with the same parameters as above. For the autoencoder, the standard deviation is calculated in the reconstruction, since the nonlinear processing of the autoencoder changes the statistics of the data. In case of the mean background subtraction, the standard deviation does not change.

Results with the Gaussian detection window for the person and drone data are shown in Figure 19 as logarithmic ROC curves together with the results from above.

The results with the Gaussian window are twofold, the detection rates for low false alarm rates are improved, but the AUC is reduced. The values of the AUC for the different cases are summarized in Table V and for a further evaluation of the results, some exemplary plots are shown in Figure 20 and 21 for a relatively weak echo of the drone data and a comparably strong echo of the person data, respectively.

TABLE V
Comparison of the AUC for backgound subtraction (BS) and autoencoder reconstruction (AE) with and without Gaussian window

	Person	Drones
AE	0.9841	0.9882
AE with Gaussian window	0.9670	0.9777
BS	0.9915	0.9906
BS with Gaussian window	0.9893	0.9790

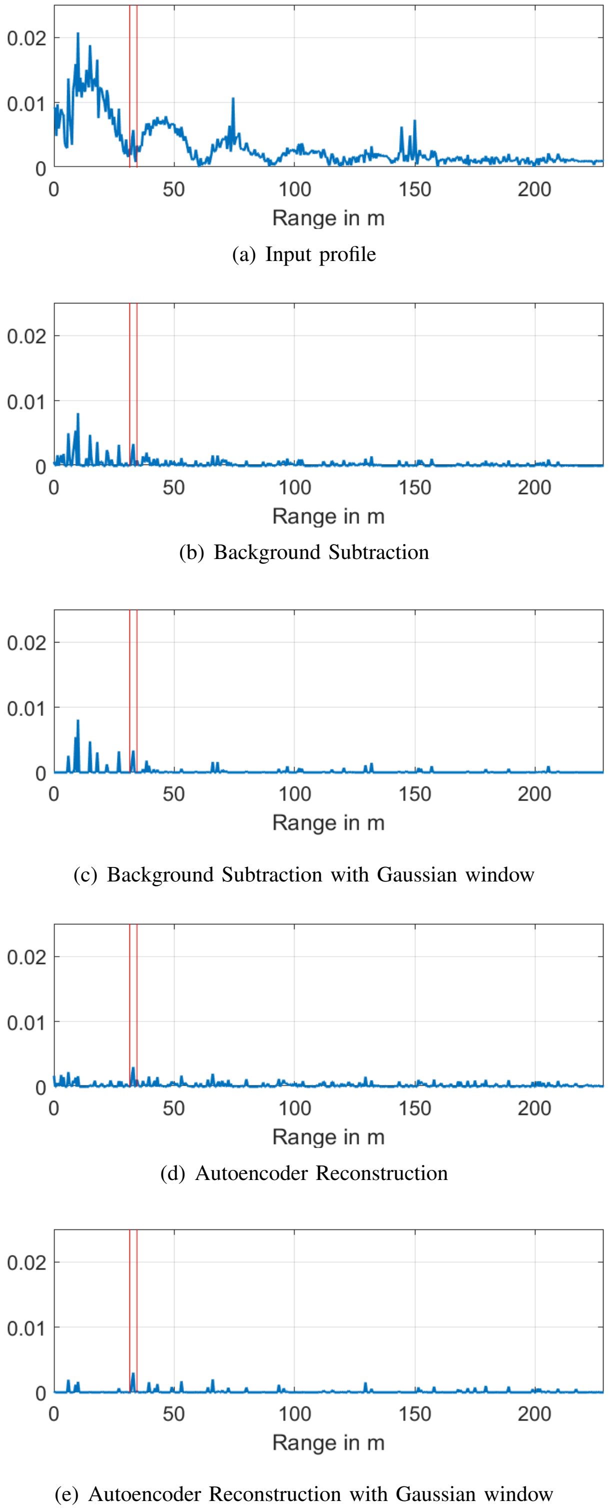


(a) Logarithmic ROC curve person

(b) Logarithmic ROC curve two drones

Fig. 19. Influence of the Gaussian detection window on the detection results

The example of the drone explains the higher number of false alarms for a given detection rate at the begin of the ROC curves. The high peaks at the begin of the profile in Figure 20 (b) are higher than the actual target between the two red lines and will trigger false alarms in the threshold detection. These high peaks remain in the profile in plot (c) of the Figure, which shows the profile of plot (b) after the application of the clutter suppression window. In the reconstruction of the autoencoder in plot (d), the actual target is the maximum of the profile, since the clutter area at the begin of the profile is already suppressed by the contractive behavior of the network. The application of the Gaussian window in the autoencoder reconstruction is shown in plot (e). This profile has the maximum still at the position of the target, but the small peaks along the remaining part of the profile are suppressed. The situation is comparable in the profile of the person in Figure 21, although the target is the actual maximum in all profiles. The very low values along the noise floor are changed to a value of zero and the higher peaks in the clutter area remain unchanged at the same position or are reduced in amplitude.



(a) Input profile

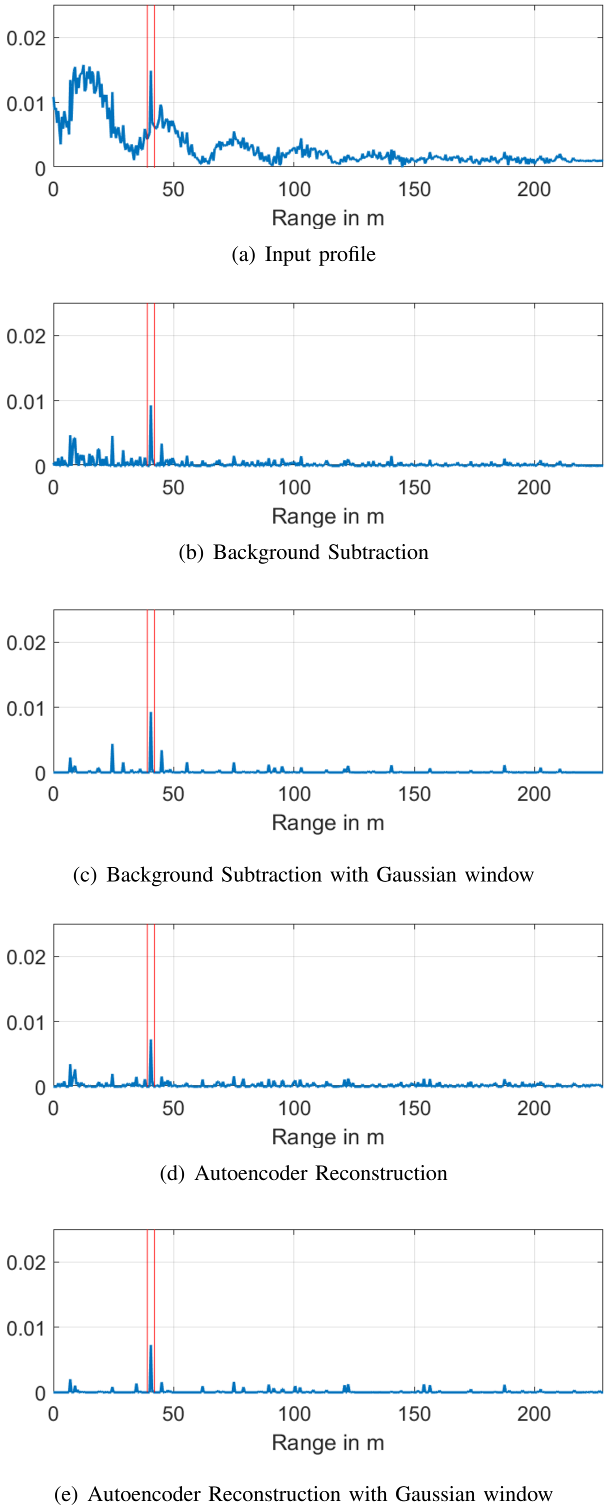
(b) Background Subtraction

(c) Background Subtraction with Gaussian window

(d) Autoencoder Reconstruction

(e) Autoencoder Reconstruction with Gaussian window

Fig. 20. Drone profile with autoencoder reconstruction and background subtraction



(a) Input profile

(b) Background Subtraction

(c) Background Subtraction with Gaussian window

(d) Autoencoder Reconstruction

(e) Autoencoder Reconstruction with Gaussian window

Fig. 21. Person profile with autoencoder reconstruction and background subtraction

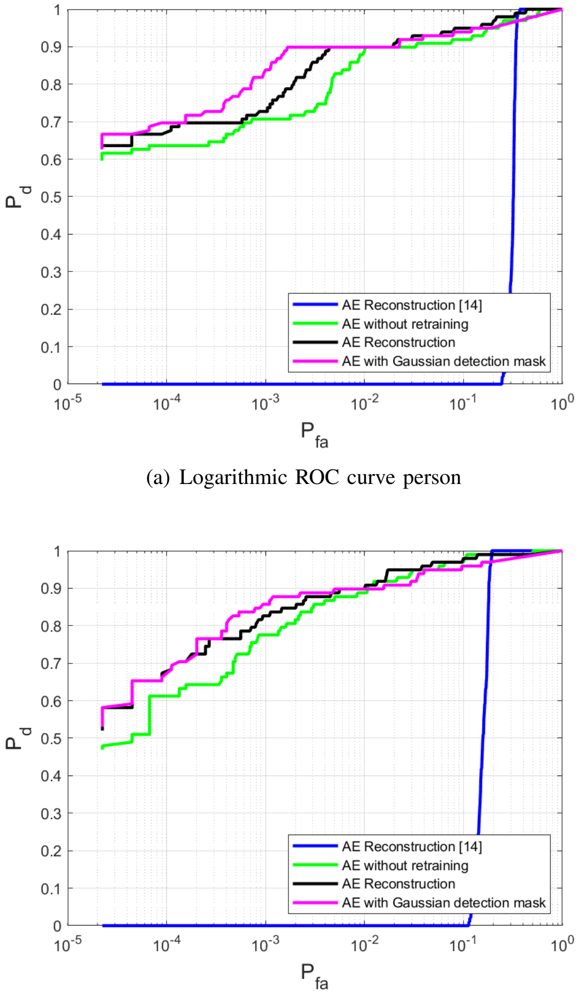
The reduction of very small values to zero is the reason for the reduction of the AUC. In case of such a weak echo that the target cannot be seen in the clutter, e.g. the second profile in Figure 9 (b), the actual target is reduced to a value of zero and is not detected anymore, even with very low thresholds. Without the application of the Gaussian window, these very low values raise the detection rate for low thresholds and thus high false alarm rates. To quantify the reduction of the clutter peaks using the Gaussian window, the SCR described above is calculated also for the data of the drones and the person without corner generated with the windows. The results are given in Table VI with the results of the autoencoder labeled with the superscript GAE and the results of the Gaussian background subtraction labeled with GBS.

TABLE VI
Comparison of the SCR for backgound subtraction (GBS) and autoencoder reconstruction (GAE) with Gaussian window

	Person	Drones
SCR^{GBS}_{min}	0.0	0.0
SCR^{GAE}_{min}	0.0	0.0
SCR^{GBS}_{max}	9.4	12.85
SCR^{GAE}_{max}	15.3	12.36
$\overline{SCR}^{GBS}$	2.58	2.52
$\overline{SCR}^{GAE}$	4.46	3.55
$\#\{y^{GBS}_{fa}\}$	29	28
$\#\{y^{GAE}_{fa}\}$	13	15
$\overline{SCR}^{GAE}_{gain}$	1.73	1.41

A comparison of Table VI and IV shows an improvement in all values except the minimum SCR, which is now 0 for all cases. This is the confirmation that in at least one case the range cell containing the target is reduced to zero. The maximum value is in case of the drone data actually higher for the Gaussian background subtraction than for the autoencoder. However, the mean value and the number of profiles with the maximum outside the target range cell is significantly higher for the autoencoder reconstruction.

To evaluate the gain in performance for the different stages of the algorithm presented here, a detection result is also shown for the autoencoder without the retraining described above. In Figure 22, the ROC curves calculated after the first training of the autoencoder is shown with the two ROC curves of the previous plots. These curves show a clear improvement, especially for low false alarm rates with the increasing training effort proposed here. The curve of the autoencoder created with the algorithm of [14] is shown only for completeness.



(a) Logarithmic ROC curve person

(b) Logarithmic ROC curve two drones

Fig. 22. ROC curves of the different autoencoder training levels

These examples show the potential of this data-driven machine learning method. With measured data, the results can be improved for low false alarm rates, which was also confirmed by the simulated data.

VI. Discussion and Conclusion

In [14], a preliminary stage of this algorithm was presented, which was intended as a proof of concept. With the measurement performed here, the method was significantly improved, allowing an increase in detection performance compared to the chosen reference methods for low false alarm rates. However, the method needs further development to bring it closer to an operational scenario. An open and important point is the created noise floor in the output profiles. These noise peaks give many false alarms for very low thresholds, while the increased SCR gives better detection rates for the more relevant low false alarm rates. Furthermore, the autoencoder itself can be improved, e.g. an overcomplete variant, i.e. an autoencoder with a code dimension larger than the input dimension, with further restrictions in the cost function would be possible. It was observed that the network structure that was used with the simulated data does not converge with the measured training data. It is assumed that the reason is the lack of training data, which allowed only the training of a very basic two-layer fully connected autoencoder network. However, the bias freezing and the retraining for denoising improved the networks detection rates that the overall performance was above the reference method for low false alarm rates.

Another possible option to improve the performance is the combination of background subtraction in areas of constant clutter and autoencoder reconstruction in areas with high variations. However, it is not possible to say how the network reacts if the target appears in these areas with high variations, since the target positions were at rather stable clutter areas during the measurements. Related to clutter suppression, the contractive behavior of the autoencoder has clear benefits in areas with high fluctuations in the background, but the higher noise floor creates more false alarms in stable background areas. This behavior should be considered in the design of a systematic approach. Furthermore, the scenario used here is, at least for a surveillance radar, a short range application. If this method should be used for example at an airport, the instrumented range must be larger and in this case it is almost impossible to measure the background without any targets. In this case, the background cannot be determined exactly and an uncertainty must be taken into account. Another open point is the adaptation to a change in the background, e.g. removed containers or trees that have been cut. At the moment, the only option is to perform a new training with new measurements of the new background. There is not yet an option for an on-the-fly adaptation to a change in the background.

These are only some of many points where the method can be further improved and therefore this method remains a part of our future research.

REFERENCES

[1] G. Lykou, D. Moustakas, and D. Gritzalis, “Defending airports from uas: A survey on cyber-attacks and counter-drone sensing technologies,” *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3537

[2] G. Fang, J. Yi, X. Wan, Y. Liu, and H. Ke, “Experimental research of multistatic passive radar with a single antenna for drone detection,” *IEEE Access*, vol. 6, pp. 33 542–33 551, 2018.

[3] M. Zywek, G. Krawczyk, and M. Malanowski, “Experimental results of drone detection using noise radar,” in *2018 19th International Radar Symposium (IRS)*, 2018, pp. 1–10.

[4] S. Zulkifli and A. Balleri, “Design and development of k-band fmcw radar for nano-drone detection,” in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–5.

[5] B. Taha and A. Shoufan, “Machine learning-based drone detection and classification: State-of-the-art in research,” *IEEE Access*, vol. 7, pp. 138 669–138 682, 2019.

[6] S. Engelbertz, C. Krebs, A. Küter, R. Herschel, R. Geschke, and D. Nüßler, “60 ghz low phase noise radar front-end design for the detection of micro drones,” in *2019 16th European Radar Conference (EuRAD)*, 2019, pp. 25–28.

[7] M. Caris, W. Johannes, S. Sieger, V. Port, and S. Stanko, “Detection of small uas with w-band radar,” in *2017 18th International Radar Symposium (IRS)*, 2017, pp. 1–6.

[8] D. Brooks, F. Barbaresco, Y. Ziani, J. Y. Schneider, and C. Adnet, “Drone recognition by micro-doppler and kinematic,” in *2020 17th European Radar Conference (EuRAD)*, 2021, pp. 42–45.

[9] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer Nature, 2018.

[10] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, “Change detection based on artificial intelligence: State-of-the-art and challenges,” *Remote Sensing*, vol. 12, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/10/1688

[11] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, “Range detection on time-domain fmcw radar signals with a deep neural network,” *IEEE Sensors Letters*, vol. 5, no. 2, pp. 1–4, 2021.

[12] M. Jarabo-Amores, M. Rosa-Zurera, R. Gil-Pita, and F. Lopez-Ferreras, “Study of two error functions to approximate the neyman–pearson detector using supervised learning machines,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4175–4181, 2009.

[13] T. Zhang, X. Zhang, J. Shi, S. Wei, J. Wang, J. Li, H. Su, and Y. Zhou, “Balance scene learning mechanism for offshore and inshore ship detection in sar images,” *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[14] S. Wagner and W. Johannes, “Target detection using autoencoders in a radar surveillance system,” in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–5.

[15] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, 2nd ed. Chapman & Hall/CRC, 2005.

[16] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[17] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, no. 1. Atlanta, Georgia, USA, 2013, p. 3.

[18] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

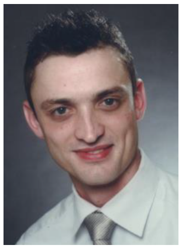
[19] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive auto-encoders: Explicit invariance during feature extraction,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML’11. USA: Omnipress, 2011, pp. 833–840.

[20] Y. Han and L. Zhu, “Improved convolutional neural network algorithm based on weight freezing method,” in *2018 24th Asia-Pacific Conference on Communications (APCC)*, 2018, pp. 341–346.

[21] M. Islam, Shahjahan, and K. Murase, “A new weight freezing method for reducing training time in designing artificial neural networks,” in *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, vol. 1, 2001, pp. 341–346 vol.1.

[22] M. Drozdov and B. Levitas, “Running gaussian average as method of adaptive background subtraction in radar applications,” in *2020 21st International Radar Symposium (IRS)*, 2020, pp. 107–111.

Simon Wagner (Member, IEEE) received the B.Eng. and the M.Sc. from the University of Applied Sciences in Trier, and his Ph.D. from the University of Siegen, Germany, all in Electrical Engineering. Before he studied, he did an apprenticeship as electronics technician for TVs and music systems. He joined the Fraunhofer FHR in 2012 for his Master Thesis and from 2013 to 2020, he has been a research scientist in the classification group of the cognitive radar department. Since 2021, he is the group leader of the machine learning for radar application group in the cognitive radar department.



Winfried Johannes obtained his diploma in electrical engineering from the Bonn-Rhein-Sieg University of Applied Sciences in 2005 and his M.Sc. in electrical engineering and information technology from the University of Hagen in 2009. Since 2005, he has been with the Fraunhofer Institute for High Frequency Physics and Radar Techniques (FHR) in the department for High Frequency Radar and Application (HRA). His main research interests include miniaturized millimeter wave radar systems for ground based and airborne applications and joint communication and radar systems.

Denisa Qosja received her B.Sc. degree in electronics and communication engineering from Epoka University in Tirana, Albania in 2018 and the M.Sc. degree in communications and multimedia engineering from FAU in Erlangen-Nuremberg, Germany in 2021. She joined the machine learning team of the cognitive radar department in Fraunhofer FHR as a research scientist in 2022. Her current research interests include target detection and classification with deep learning.

Stefan Brüggenwirth (Senior Member, IEEE) received a Ph.D. degree from the faculty of aerospace engineering at the Bundeswehr University Munich, Neubiberg, Germany in 2015. and a diploma in aerospace engineering with specialization in computer science and space systems from the University of Stuttgart, Germany, in 2008. He spend 2.5 years at NASA Ames research center and joined the Fraunhofer Institute for High Frequency Physics and Radar Techniques as head of the Cognitive Radar department in 2014. His research focus lies in NCTR, machine learning and advanced signal processing for cognitive radar applications. He is a IEEE senior member and NATO SET Panel member at large for machine learning and artificial intelligence in RF sensors.