IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 September 2022, accepted 20 October 2022, date of publication 31 October 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218349

RESEARCH ARTICLE

Universal Adversarial Attacks on the Raw Data From a Frequency Modulated Continuous Wave Radar

JAKOB VALTL[1] AND VADIM ISSAKOV[2], (Senior Member, IEEE)
[1]Infineon Technologies AG, 85579 Neubiberg, Germany
[2]Technische Universität Braunschweig, 38106 Braunschweig, Germany

Corresponding author: Vadim Issakov (v.issakov@tu-braunschweig.de)

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) in collaboration between the Projects ''KI-Flex'' within the Founding Program Microelectronic from Germany Innovation Driver under Project 16ES1027, in part by the ''KoSi'' Project funded by the German Federal Ministry for Digital and Transport (BMDV) within the Funding Program ''Automatisiertes, Vernetztes Fahren'' under Project 01MM20011C, and in part by the Project ''TEACHING'' founded by the Horizon 2020 Program under Project 871385.

ABSTRACT As more and more applications rely on Artificial Intelligence (AI), it is inevitable to explore the associated safety and security risks, especially for sensitive applications where physical integrity is at risk. One of the most interesting challenges that come with AI are adversarial attacks being a well-researched problem in the visual domain, where a small change in the input data can cause the Neural Network (NN) to make an incorrect prediction. In the radar domain, AI is not that widespread yet but the results that AI applications produce are very promising, which is why more and more applications based on it are being used. This work presents three possible attack methods that are particularly suitable for the radar domain. The developed algorithms generate universal adversarial attack patches for all sorts of radar applications based on NN. The main goal of the algorithms, apart from the computation of universal patches, is the identification of sensitive areas in the raw radar data input which than can be examined more closely. To the best of our knowledge, this is the first work that deals with calculating universal patches on raw radar data, which is of great importance especially for interference analysis. The developed algorithms have been verified on two data sets. One in the field of autonomous driving where the attacks lead to a steering misprediction of up to 0.3 for the steering value which is within [-1,1], with the results also being successfully tested on a demonstrator. The other data set originated from a gesture recognition task, where the attacks decreased the accuracy, originally at 97.0% up to a minimum of 16.5%, which is slightly above 12.5% being the accuracy for a purely random prediction.

INDEX TERMS Adversarial attacks, artificial neural networks, autonomous vehicles, edge computing, object recognition, radar applications, real-time systems.

I. INTRODUCTION

There are many reasons why artificial intelligence (AI) will play an increasingly important role in the future. One of the main drivers is without question the transportation sector [1] as one of the central components of today's society and one of the main drivers of intensive research. In this context, numerous future aspirations, like decarbonization or time saving, are often projected into a transformation of the transport structure. The promise is that thanks to AI, autonomous vehicles will be able to drive on the roads of the future and, due to optimized efficiency, will not only get us to our destination faster and with fewer resources, but also reduce the overall volume of traffic by means of sharing concepts of autonomous self driving cars [2], [3].

Typically, the algorithms that enable the autonomous cars make their decisions based on data provided to them by sensors, which in turn record the vehicle's environment. On the

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

technical side there are several reasons in favour of AI in the field of autonomous driving. On the one hand, tough restrictions on the real-time capabilities of algorithms, that can be met with AI, play a particularly important role so that certain reaction times can be guaranteed, and on the other hand, the high data rate and the associated data processing speak for the use of AI because it works extremely efficient and quickly on specialized hardware components.

However, this transformation in the transport sector towards AI can only succeed if the vehicles of tomorrow meet safety requirements. In particular, it must be ensured that the AI algorithms are not only capable of recognizing traffic situations under laboratory conditions and acting accordingly, but also function flawlessly under adverse conditions [4].

The constantly accelerating development of AI algorithms and the incredible precision with which predictions can be made do not disguise two limiting factors. These are that, especially in the case of complex networks, explainability is not given and decisions can be significantly influenced by means of adversarial attacks, that are targeted attacks that maliciously and intentionally cause AI in autonomous vehicles or subsystems to behave incorrectly [5]. Adversarial attacks lead to an erroneous prediction of an artificial neural network (ANN) when the input signal has been minimally altered, to provoke this erroneous prediction. This is of particular interest as it is a security risk that it is often not recognized by humans, or at least not perceived as an attack as it is hardly noticeable and exclusively affects the AI algorithms. Nonetheless these attacks must be detected and thwarted.

In an autonomous vehicle, a variety of different sensors will be used, such as camera, lidar/time-of-flight camera, ultra-sound and radar. One of the reasons for the detailed study of AI algorithms in the radar domain is the particular prominence of this sensor type in contrast to other available sensors. Radar is expected to play a leading role due to its independence from weather conditions such as visibility obstructions caused by fog, snow or rain [6], [7]. In contrast to lidar, radar also offers the advantages of having no mechanically moving components, a faster repetition rate, the ability to detect obscured objects indirectly by means of second-degree reflection, and the ability to directly determine the radial velocity of objects by means of the Doppler effect. Each sensor component and the associated evaluation algorithms must be tested separately for sensitivity to attacks such as adversarial attacks.

This work focuses on the creation of universal adversarial attack patches for radar data evaluation algorithms. With the setting of the boundary conditions the characteristics of the attack, patch vs. perturbation, can be steered. Particular emphasis was placed on the feasibility of real world implementation. The created tools can be used to analyse the sensible areas in the radar signal, to determine where the weak points of AI algorithms in raw radar data processing are. It is important to note that research on adversarial attacks should not be seen exclusively as an attack strategy. Rather, adversarial attacks can be defined as standards to certify the resilience of AI and are thus an ideal tool for standardization and thus qualification and certification [8].

II. BASICS AND PRIOR WORK

A. RADAR TECHNOLOGY

The development of radar technology can look back on a long history [7]. Nowadays, in consumer electronics as well as in the automotive sector, the frequency modulated continuous wave (FMCW) technology is the one most widely used. For each frame a packet of chirps is transmitted, where a chirp is a frequency modulated signal defined by its bandwidth and length. The reflections of the chirps are detected by receiving antennas and together withe the original transmitted signal, the intermediate frequency signal is computed. This resulting signal is sampled to produce a 3 dimensional data packet per frame defined by a) the number of antennas, b) the number of chirps per frame (CPF) and c) the number of samples per chirp (SPC). High center frequencies allow a fine resolution of the detected environment [9], [10]. The advantages of radar technology in general and FMCW in particular are on the technical side, the precise detection of static objects and the determination of the relative speed of moving objects, as well as the guarantee of privacy on the data protection side.

B. AI

Due to the fact that ANNs are increasingly better studied, which is reflected in an increasingly higher accuracy of the prediction probability and can cope with a high data throughput, thanks to specialized hardware, there is a trend that the processing of radar data will increasingly be done using ANNs. In previous approaches, radar data is usually preprocessed before the information is transferred to an ANN in the form of Range-Doppler-Maps (RDMs) or other representations. Various applications of radar sensors are conceivable, ranging from distance determination and classification of objects in road traffic [11], to gesture recognition in consumer electronics [12] and even breath and heartbeat frequency detection of humans [13]. There is also an increasing trend to move the previous data preprocessing steps to the ANN as well, [14] as this allows specialized AI accelerators to shorten response times and enable processing on the edge [15], [16]. In previous work it was shown that this approach is in no way inferior to previous, traditional methods if sufficient training data is available [17].

C. ADVERSARIAL ATTACKS

Especially in the visual domain there is a lot of research work dealing with the problem of adversarial attacks. Usually, the goal of adversarial attacks is to provoke a malfunction by changing the input data of the ANNs as little as possible. On recorded datasets, this criterion seems reasonable, as it is counter intuitive especially for humans, as external observers, and shows limitations of AI. Basically, adversarial attacks have to be distinguished between live attacks and attacks on recorded data sets. While the latter usually only reveals the

existence of vulnerabilities of the AI, live attacks represent a real danger for AI systems. The criterion that an adversarial attack is specified by a minimal change of input data is in this case no longer relevant, since usually a control instance such as a human is not involved in the decision cycle. Instead, it is important to find attacks that are as effective as possible and deliver falsified but still plausible sensor data. The main focus is the optimization for the enforceability of the disturbance in live attacks. Impressive examples of live attacks in different sensor domains are the work of [18], [19], and [20], where a stop sign is no longer recognized as such, or an object on the road is overlooked due to its surface structure / shape and even active spoofing is investigated.

In the radar domain, research has not progressed that far, and in addition, preprocessing of radar data complicates attack methodologies. Thus, there is some work in the radar adversarial attack area, but their commonality is the limitation to attack scenarios on already preprocessed and recorded data [21]. This means that the radar-specific signal shape is not taken into account, but the problem is transformed into a visual representation and attacked with successful algorithms from the visual domain. A further sticking point of the previous work is the limitation on frame-wise creation of attack masks. In the visual domain, several works already exist on universal patterns that produce malfunction of ANNs independent of the sensor input signal [22]. These patterns can be computed a priori, so that no computational effort is required to generate them during the attack. The previous work in the radar domain pays particular attention to keeping the intensity of the mask as small as possible when creating the masks in the Range Doppler domain. This makes sense when investigating the problem with tools from the visual domain to assure that adversarial attacks on radar data exist in principle, but are not useful for performing live attacks or determining the probability of such attacks in the radar domain as it is neglecting the preprocessing pipeline.

In this work, the latter two challenges are addressed by developing several algorithms that compute universal patterns that can ideally be generated in reality. For the network structure to be attacked, preprocessing is omitted in order to impose constraints on the mask with respect to the feasibility of these patches in the analog world. Due to the transferability known in other areas, it can be assumed that successful attacks will not be limited to the architecture on which they are generated but will also have an effect on unknown ANNs [23].

III. METHODOLOGY AND ALGORITHMS

The developed algorithms do not intend to compute adversarial patterns that stick out for their small perturbation. The focus rather lies on the generation of patches that are targeted and universal, thus independent on the legitimate input. This will lead to a failure of the network output, most of the time. Another key focus is the feasibility of applying them in the analog world. This approach allows to investigate real world thread scenarios.

In order to be as close as possible to the real world perception, which is what the sensor itself detects, we take the raw data, coming from the analog digital converter (ADC), as input to our algorithms. This way we do not loose information that might get lost during traditional preprocessing steps. As the preprocessing itself does not provide additional information but only consists of filtering and representation shifts, we presume that this can be handled by the ANN itself. This claim was proven in our previous work [17], showing that not only networks of variational autoencoders are capable of accomplishing that.

The raw radar sensor signal coming from the ADC is usually quantized as a 12 bits float within 0 and 1. This signal is used as an input for the ANN.

The two task types that are taken into account are:

- Classification like in gesture recognition and
- Regression as in angle detection.

Depending on the task the system is supposed to solve, the output of the ANN is either a distribution on the predicted classes or floating values representing the output.

The goal is either to output a wrong class as a prediction or to deflect the output value in a certain direction.

The algorithms are designed as a white box attacks though if the ANN is unknown, a sufficiently similar network can be generated and attacked with this algorithm and due to the effect of adversarial patterns being transferable, the unknown ANN can be attacked.

All implemented algorithms are based on the gradient method which is also used for training ANNs, but instead of the weights' gradient the gradients of the input neurons to the output neurons are considered. Depending on the chosen attack method, different parts of the gradients, possibly taking into account distance weightings, are combined to form a universal adversarial patch.

During the computation of the gradient the loss function must be adjusted. In the case of a regression network, the output of the network is used directly without considering the labels of the samples. If a classification network is present, the labels are also not taken into account, but the outputs of the individual classes of the network. The patch for each individual sample is calculated by subtracting the gradient of the predicted class from the gradient of the targeted class taking into account the predictions.

The following subsections describe the different universal adversarial attack algorithms that we developed.

A. ADDITIONAL INPUT-BIAS LAYER

This method is by far the simplest way to generate a universal attack, as it just introduces an Additional Input-Bias Layer (ABL). In front of the already trained network, an additional bios layer is put on the input layer. Then the loss function of the network is adapted depending on the task type while all pre-trained weights of the network are frozen. Training the network only changes the additionally added bias layer. This

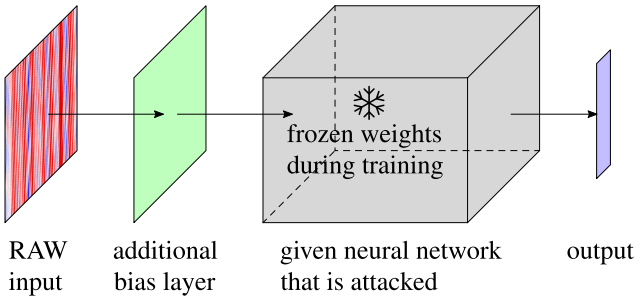

FIGURE 1. Architecture of the additional input-bias-layer attack.

bias layer represents the universal adversarial patch. A visual description is given in Fig. 1.

Deviating with regard to the above explanation on the loss function the same is retained in the case of a classifier and only the target class is specified as label class during computation of a universal patch.

This method also easily permits the application of various conditions on the patch. Limits of values or empty spaces in the patch can be set. This can a priori ensure the later applicability of the patch by, for example, external sources of interference.

B. SUMMATION OF PATCHES

For this algorithm all gradients of the individual samples are computed on the pre-trained network that is attacked. From the gradients the patches are generated, either by taking the gradient directly or combining them with respect to the corresponding network output. The summation and normalization of all generated patches represent the universal patch generated by this method, hence Summation of Patches (SoP).

Algorithm 1 Overall Algorithm With a Focus on the Computation of the Framewise Adversarial Patches P

Require: X, Y, T, f, g
1: $u \leftarrow 0$
2: **for** all epochs **do**
3: $\quad X \leftarrow X + u$
4: $\quad$ **Evaluate** with gradient tab
5: $\quad\quad \hat{Y} \leftarrow f(X)$
6: $\quad P_Y \leftarrow \nabla(\hat{Y}_Y, X)$
7: $\quad P_T \leftarrow \nabla(\hat{Y}_T, X)$ ▷ Is zero in case of regression NN
8: $\quad P \leftarrow k \cdot P_{\hat{Y}} - (1-k) \cdot P_T$
9: $\quad u \leftarrow g(u, P, \ldots)$ ▷ g is update of the universal patch
10: $\quad$ save u
11: **end for**

The algorithms SoP and WSoP both have the general algorithm structure in common. Algorithm 1 describes this overall architecture of the gradient approach where X is the data that gets evaluated by the ANN $f()$ resulting in the prediction $\hat{Y}$. Y is the correct label, T is the target class in case of a classification problem. The universal patch u gets updated by g which is described in Algorithm 2 taking the framewise adversarial patches P as an input. P_y and P_t are the gradients with respect to the output of the subscript, while k balances between the addition of the target and the subtraction of the label gradient. Line 7 in Algorithm 1 only applies in case of a classification problem, otherwise it is a zero matrix.

Algorithm 2 Universal Patch on Basis of Summation

Require: $u, P, \alpha, \epsilon, m$
Ensure: $u \in m$
1: $\hat{u} \leftarrow \sum P$
2: $u \leftarrow (1-\alpha) \cdot \|u\|^2 + \alpha \cdot \left\|\hat{u}\right\|^2$
3: $u \leftarrow \epsilon \cdot u$
4: **if** $u \notin m$ **then**
5: $\quad$ set u to the constraints of m
6: **end if**
7: Return u

Algorithm 2 explains the updating process of the universal patch on the basis of summation, following the same nomenclature as before. The additional variables that are not explained in Algorithm 1 are, α the update weight of the universal patch for each epoch, ϵ the intensity of the patch and m the feasibility mask to constrain the patch.

C. WEIGHTED SUMMATION OF PATCHES

Similar to the previous explained *SoP* algorithm, all individual patches are computed on each individual sample as described in Algorithm 1. But instead of a pure summation, each patch is multiplied with a weight that represents the importance of each individual patch. There are different weighting methods. The goal is for the universal patch to cause a similarly large and evenly distributed shift on all samples. For this reason a weight factor is added resulting in a Weighted Summation of Patches (WSoP).

To achieve this, the weighting in the case of regression networks takes into account the distance of the prediction to the actual label. In the case of classification networks the emphasis each patch is given is determined by the difference of probability between the targeted class and the predicted one. The following formula represents the weight function,

$$w = s^{-d}, \tag{1}$$

where w is the weight, s is a scalar hyperparameter, determined experimentally and d is either the difference of prediction and label or the difference of the class-probabilities.

Conditions on the values of the patch can also be set. But for ensuring the gradient to consider those it is needed to repeat the algorithm for various epochs while the universal patch is applied.

IV. EXPERIMENTAL RESULTS

The described algorithms are verified with two recorded data sets.

Algorithm 3 Universal Patch on Weighted Summation

Require: $u, P, Y, T, \alpha, \epsilon, m$
Ensure: $u \in m$
1: $d \leftarrow \hat{Y}_Y - \hat{Y}_T$
2: $\hat{u} \leftarrow \sum P \cdot w$
3: $u \leftarrow (1-\alpha) \cdot \|u\|^2 + \alpha \cdot \|\hat{u}\|^2$
4: $u \leftarrow \epsilon \cdot u$
5: **if** $u \notin m$ **then**
6: set u to the constraints of m
7: **end if**
8: Return u

TABLE 1. Radar sensor operation parameters for the regression task.

Parameter	Value
center frequency	60 GHz
bandwidth	[58 – 62] → 4 GHz
ADC sampling frequency	2 MSps
chirps per frame	64
chirp time duration	112 µs
samples per chirp	128
frames per second	30

A. DATASET DESCRIPTION

To investigate the two problem classes, regression and classification, two different tasks were investigated. The sensor used for the recording is the same in each case. It is the FMCW radar sensor from Infineon BGT60TR13C which has a center frequency of 60 GHz. To cope with sensor specific requirements the sensordata undergoes a minor preprocessing in the form of a baseline removal and a gain-correction. The two data sets are structured as follows:

1) REGRESSION

The task is taken from the automotive sector and describes the following scenario: A model car equipped with a radar sensor follows another model car autonomously. This maneuver is called platooning. The control of the pursuing car is taken over by an ANN. Details about the experimental setup can be taken from our previous works [24], [25].

For this task a data set was gathered, consisting of 153 648 radar frames and the associated driving parameters that are used as labels. The individual frames contain the data from two antennas and were recorded with 64 chirps per frame (CPF) and 128 samples per chirp (SPC). The dataset was split as such that, 103 899 frames are used for training, 34 634 for evaluation and finally 15 115 for testing. Only the steering position is taken into account as a performance parameter of the NN, as this is the most difficult parameter to determine, and also from a safety aspect, the most critical one. The operation parameters of the radar sensor are given in Table 1.

2) CLASSIFICATION

The classification data set comes from the field of human gesture recognition and distinguishes between 8 hand gestures:

1) class_0: down_up
2) class_1: up_down

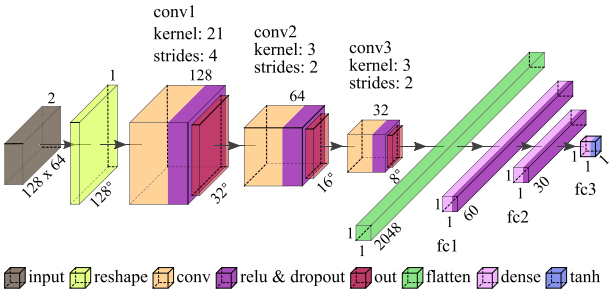

FIGURE 2. Architecture of attacked regression network.

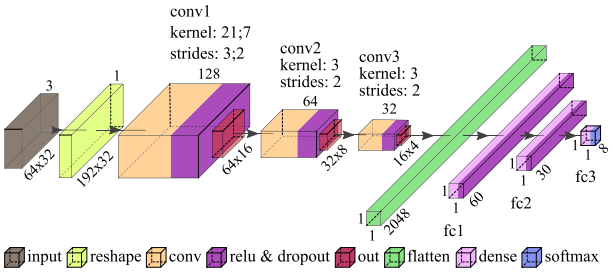

FIGURE 3. Architecture of attacked classification network.

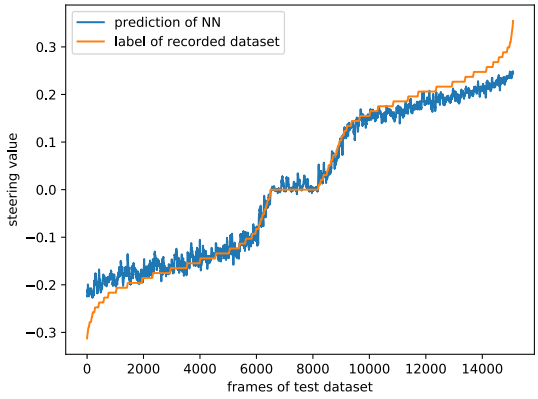

FIGURE 4. Results of regression on test dataset.

3) class_2: left_right
4) class_3: rubbing
5) class_4: right_left
6) class_5: diagonal_sw-ne
7) class_6: diagonal_se-nw
8) class_7: clapping

The dataset is a subset of the one presented by Chmurski [12]. Each gesture consists of 10 frames with the information of 3 antennas recording with 64 CPF and 32 SPC. Each gesture was repeated 570 times which results in a data set consisting of 45 600 frames. The entire dataset is split such that 342 repetitions of each gesture are used for training, and 114 each for validation and testing. The split is kept constant for all parts (network training and attack computations). The operation parameters of the radar sensor are given in Table 2.

B. NETWORK ARCHITECTURE AND TRAINING

Details about the neural networks, handling the tasks are given below.

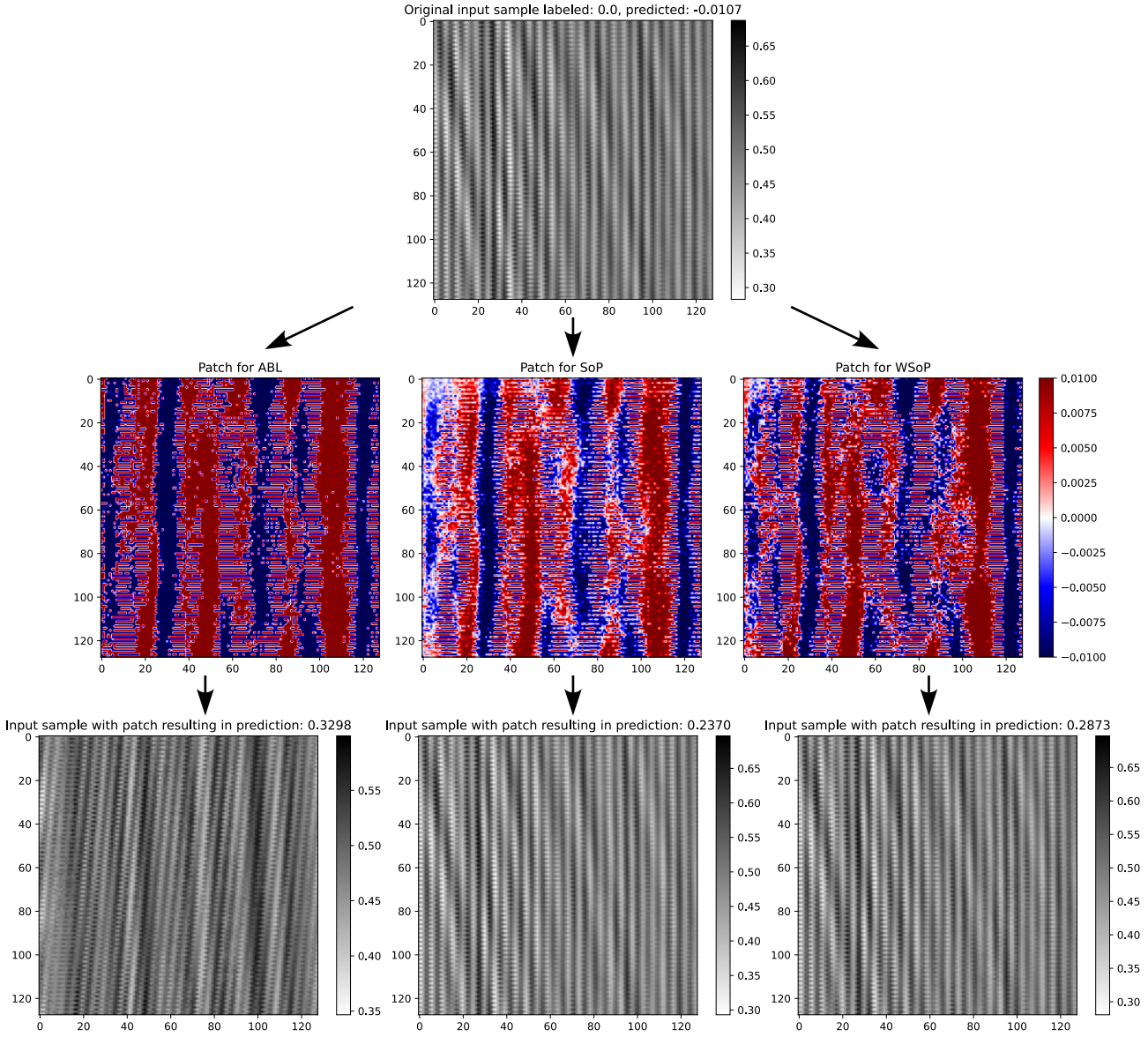

FIGURE 5. Patches on a single sample.

TABLE 2. Radar sensor operation parameters for the classification task.

Parameter	Value
center frequency	60 GHz
bandwidth	$[57-63] \rightarrow 6$ GHz
ADC sampling frequency	4 MSps
chirps per frame	64
chirp time duration	37 µs
samples per chirp	32
frames per second	10

TABLE 3. Hyperparameters for training the networks. Parameters that are the same for both tasks are aligned in the center.

Parameter	Value for regression	Value for classification
dropout	0.05	
activation	relu | tanh	relu | softmax
optimizer	Adam	
learning rate	10^{-5}	10^{-4}
loss	mean squared error	categorical crossentropy
batch size	500	
max epochs	200	
early stopping	true	
patience	5	
decay	10^{-10}	

1) REGRESSION

For the regression problem a network was trained which is described in more detail in Fig. 2. The three dimensional data packages for each frame are concatenated to two dimensional ones. This way each frame has the shape 128 × 128x1. The network is trained with the hyperparameters given in Table 3, and early stopping is triggered as soon as the MSE does not decrease more than 10^{-10} for 5 consecutive epochs. Each convolutional- as well as fully connected layer (except for last) is followed by a ReLU and dropout layer.

TABLE 4. Hyperparameters for attacking the regression task.

Parameter	ABL	SoP	WSoP
max epochs		30	
m clip limits		[-0.01; 0.01]	
optimizer	Adam	–	–
learning rate	10^{-4}	–	–
batch size		512	
α	–	1/epoch	1/epoch
ε		0.02	
s	–	–	0.2

2) CLASSIFICATION

Gesture recognition is performed using the network shown in Fig. 3. Each frame is examined individually and then the class that received the most votes among the 10 frames is the final prediction. The stacking of the 3 dimensional data packages is handled similar to the one of the regression task. Though due to the different set of CPF and SPC the final input dimension after stacking has the shape 192×32.

C. PERFORMANCE OF THE ATTACK ALGORITHMS

In the following, the performance of the presented attacks will be examined, first the behavior without attack mask will be presented and later on compared with the behavior under the different attack conditions. All results shown were obtained on the test dataset.

1) REGRESSION

Without the attack pattern, the deviation of the output of the neural network with respect to the recorded steering position is on average 0.028. Figure 4 shows the output of the neural network and the corresponding label. The individual frames on the x axis are sorted in ascending order according to the label value.

The calculation of the masks is carried out according to the schemes given above, using the parameters given in Table 4:

In Fig. 5 on the next but one page, the different patches of the algorithms are displayed, exemplary for one run on a single sample.

Figure 6 top diagram shows the relationship between epsilon, generated errors and the number of epochs. The picture is from the ABL attack on the regression task and is exemplary for the other attacks as well. In principle, the generated error increases with increasing epsilon and increasing epoch number. Figure 6 bottom shows the distribution over the individual samples for different epsilon values, each for 30 epochs of training. It is to be noted that with a right drift as an attack target especially the samples during the left drift are influenced. The same is true for the other way around. For better comparability, all algorithms are run with an epsilon of 0.02 and 30 epochs. When the attack patterns are applied under these conditions, the average steering deviation increases from the 0.028 to 0.291, 0.284, and 0.307 for ABL, SoP, WSoP, respectively. The detailed results are shown in Fig. 7 The performance of the attacks highly depends on the chosen hyperparameters, as the two example of the WSoP in Fig. 8 show. Where one of the two is optimized for maximum

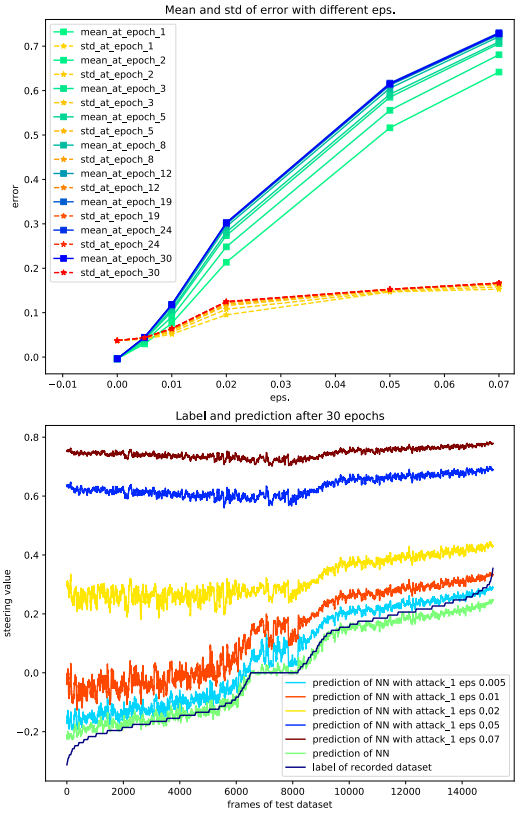

FIGURE 6. Influences on the attack performance.

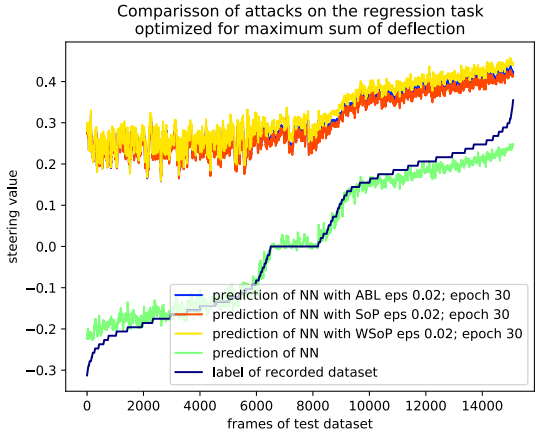

FIGURE 7. Comparison of attack methods.

sum of deflection and the other for the most uniform deflection possible.

2) CLASSIFICATION

Without any attacks, the neural network achieves an accuracy of 97.0% on the test data set where Table 5 reflects the confusion matrix of the evaluation. It can be clearly seen that all classes are recognized about equally well and none is preferred by the network.

In the further course the attack masks with an epsilon of 0.02 are added. Figure 9 shows an example of the evolution of the accuracy on the validation data set when the

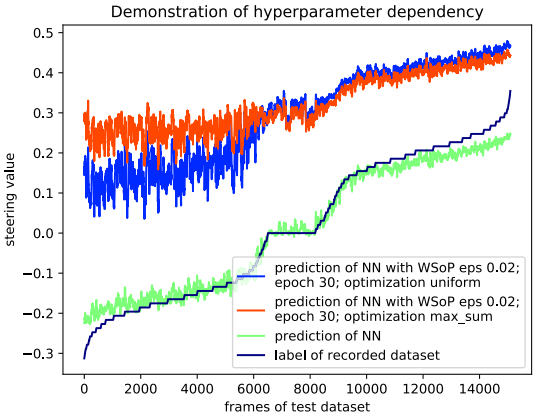

FIGURE 8. Dependency on hyperparameters.

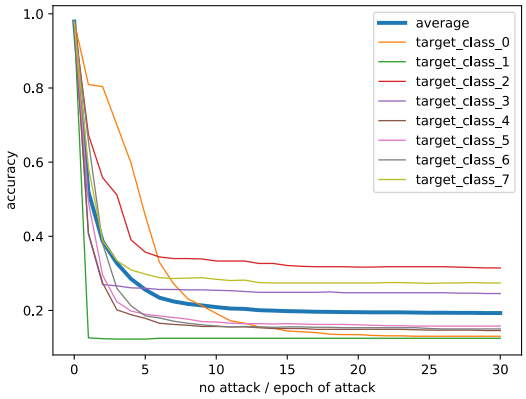

FIGURE 9. Evolution of accuracy for ABL.

TABLE 5. Confusion matrix.

Actual Classes \ Predicted Classes	0	1	2	3	4	5	6	7
0	99 87%	15 13%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
1	1 1%	113 99%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
2	0 0%	0 0%	114 100%	0 0%	0 0%	0 0%	0 0%	0 0%
3	0 0%	0 0%	1 1%	113 99%	0 0%	0 0%	0 0%	0 0%
4	0 0%	0 0%	0 0%	0 0%	109 96%	2 2%	2 2%	1 1%
5	1 1%	0 0%	0 0%	0 0%	0 0%	112 98%	1 1%	0 0%
6	0 0%	0 0%	0 0%	2 2%	0 0%	1 1%	111 97%	0 0%
7	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	0 0%	114 100%

TABLE 6. Confusion matrix.

Actual Classes \ Predicted Classes	0	1	2	3	4	5	6	7
0	13 11%	2 2%	0 0%	0 0%	2 2%	93 82%	4 4%	0 0%
1	10 9%	52 46%	0 0%	0 0%	0 0%	51 45%	1 1%	0 0%
2	0 0%	0 0%	1 1%	0 0%	0 0%	113 99%	0 0%	0 0%
3	0 0%	0 0%	0 0%	2 2%	0 0%	109 96%	3 3%	0 0%
4	0 0%	0 0%	0 0%	0 0%	0 0%	114 100%	0 0%	0 0%
5	0 0%	0 0%	0 0%	0 0%	0 0%	114 100%	0 0%	0 0%
6	0 0%	0 0%	0 0%	0 0%	0 0%	114 100%	0 0%	0 0%
7	0 0%	0 0%	0 0%	0 0%	0 0%	114 100%	0 0%	0 0%

attack is performed with the ABL. The attack performs best, on the unseen validation set, after 30 epochs, which was the maximum amount of epochs for which we computed the values, though it already reaches a kind of steady state after 20 epochs.

As an example, the target class number five in this take at epoch 30, is examined more closely. Its confusion matrix on the test dataset is the one shown in Table 6, showing that some classes are harder to fool than others, mostly depending on the similarity between gestures. In the confusion matrix, for example, it is clear that the gestures of class zero and one are very similar to each other, but very different from the gesture of class five. Figure 10 on the next page shows one computed patch for each attack, that is applied to a single frame of a gesture of class 3, converting it into a class 5 prediction. A frame that actually is belonging to class 5 is given in Fig. 11. Similar as in the regression task the weights of the ABL attack are more at the limits of ϵ.

The following results are all from the patches computed for 30 epochs. Averaged over all classes and three separate takes, the attacks lead to the fact that in 87.7%, 66.5%, and 89.0%, (for ABL, SoP, WSoP) of all cases a previously correctly classified sample of a non-target class is wrongly attributed to the target class, from now on we call this parameter the attack rate. The accuracy thus decreases to 19.3%, 21.1%, and 16.5%, whereby it must be noted that the accuracy alone only reflects the effectiveness of the attacks to a limited extent, since even if all samples are assigned to the target class, the accuracy is still $1/amount\_of\_classes$, assuming that the number of samples of all classes is the same. But the far greater disadvantage of the accuracy is that it only reflects how much the correct prediction has deteriorated without, however, allowing a statement as to whether the attackers target class has been reached. The attack rate pays particular attention to that. Figure 12 shows the evolution of the attack rate, as well as the accuracy always for one take for all three attack algorithms.

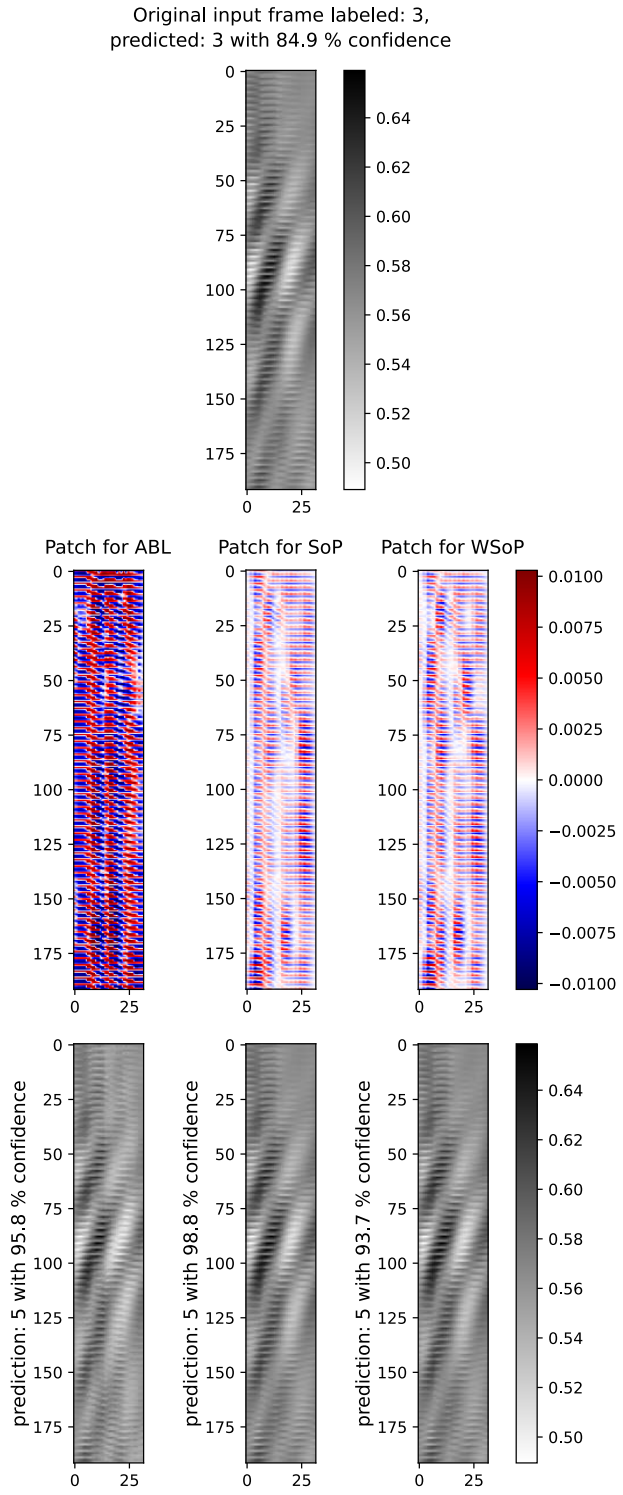

FIGURE 10. Example of patches for each attack, applied to a single frame of class 3 and the change of prediction.

The parameters for the attack where chosen as displayed in Table 7

3) COMPARISON

ABL pushes more towards the limits which can be seen in Fig. 5 and Fig. 10 while (W)SoP generates patches that have a higher variation of disturbance values. The statement is true in both cases. (classification and regression) Table 8 is a summary of the main results of both tasks over all attack methods.

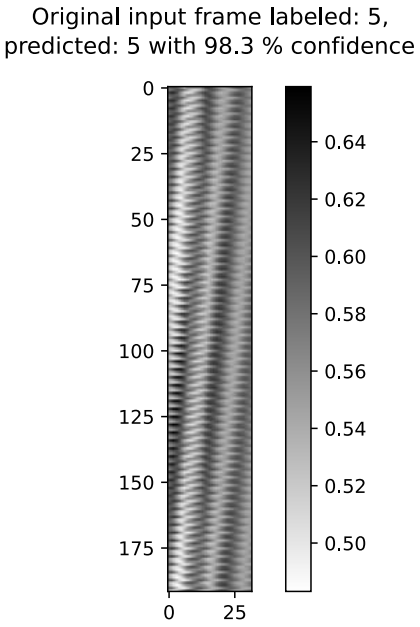

FIGURE 11. Example a single frame of class 5.

TABLE 7. Hyperparameters for attacking the classification task.

Parameter	ABL	SoP	WSoP
max epochs	30		
m clip limits	[-0.01; 0.01]		
optimizer	Adam	–	–
learning rate	$2 \cdot 10^{-5}$	–	–
batch size	500 frames (50 samples)		
α	–	1/epoch	1/epoch
ε	0.02		
s	–	–	0.15

TABLE 8. Summary of main results.

	no attack	ABL	SoP	WSoP
deviation of prediction to label for the regression task	0.028	0.291	0.284	0.307
accuracy for the classification task	97.0%	19.3%	21.1%	16.5%
attack rate for the classification task	–	87.7%	66.5%	89.0%

V. DISCUSSION AND CONCLUSION

Regarding the regression problem, all attacks are sufficient to make the system stop working. However, the WSoP attack offers the best possibilities to adapt the attack to the respective attack targets and possible available interference patterns. On the other hand the WSoP turned out to be the most challenging attack regarding fine tuning of the hyperparameters. On the classification problem we observed that not all classes can be attacked equally well. All attacks managed to decrease the original accuracy significantly well but

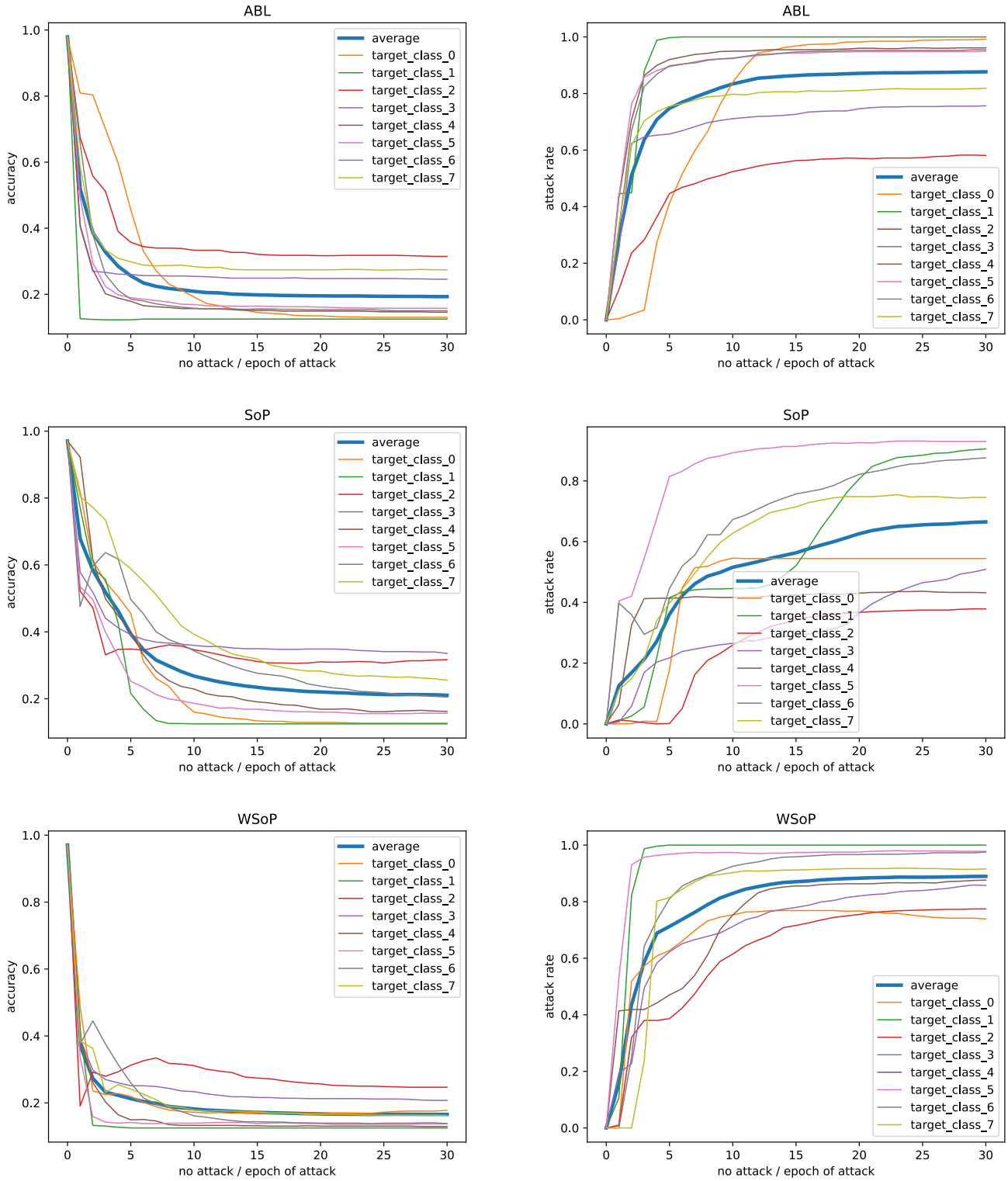

FIGURE 12. Evolution of accuracy and attack rate for all attack methods.

actually fooling the network in the prediction of a target class varies. The experimental results show, that targeted universal attacks can be generated on raw radar data, even with boundary conditions that restrict the attack pattern. This way the developed algorithms are a tool to compute attack patterns on the basis of interference behaviour or other disturbance mechanisms.

The next steps towards a live attack are investigations into the interference behaviour to detect vulnerable regions in the raw data and exploit them. Additionally safety risks of

unintended interference due to multiple traffic participants can be evaluated, which probably will be the most relevant future research topic.

REFERENCES

[1] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/1/189

[2] M. Zhu, X.-Y. Liu, F. Tang, M. Qiu, R. Shen, W. Shu, and M.-Y. Wu, "Public vehicles for future urban transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3344–3353, Dec. 2016.

[3] A. Sumalee and H. W. Ho, "Smarter and more connected: Future intelligent transportation system," *IATSS Res.*, vol. 42, no. 2, pp. 67–71, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0386111218300396

[4] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable artificial intelligence for autonomous driving: An overview and guide for future research directions," 2021, *arXiv:2112.11561*.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[6] N. Scheiner, F. Weishaupt, J. F. Tilly, and J. Dickmann, "New challenges for deep neural networks in automotive radar perception," in *Automatisiertes Fahren*, T. Bertram, Ed. Wiesbaden, Germany: Springer, 2021, pp. 165–182.

[7] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar—From first efforts to future systems," *IEEE J. Microw.*, vol. 1, no. 1, pp. 135–148, Jan. 2021.

[8] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: A standardized adversarial robustness benchmark," 2020, *arXiv:2010.09670*.

[9] V. Lammert, S. Achatz, R. Weigel, and V. Issakov, "A 122 GHZ ISM-band FMCW radar transceiver," in *Proc. German Microw. Conf. (GeMiC)*, 2020, pp. 96–99.

[10] V. Issakov, A. Bilato, V. Kurz, D. Englisch, and A. Geiselbrechtinger, "A highly integrated D-band multi-channel transceiver chip for radar applications," in *Proc. IEEE BiCMOS Compound Semicond. Integr. Circuits Technol. Symp. (BCICTS)*, Nov. 2019, pp. 1–4.

[11] Z. Guo, W. Yi, Y. Wu, and T. Luo, "Robust radar detection and classification of traffic vehicles based on anchor-free CenterNet," in *Proc. Int. Conf. U. K.-China Emerg. Technol. (UCET)*, Nov. 2021, pp. 252–257.

[12] M. Chmurski, G. Mauro, A. Santra, M. Zubert, and G. Dagasan, "Highly-optimized radar-based gesture recognition system with depthwise expansion module," *Sensors*, vol. 21, no. 21, p. 7298, Nov. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/21/7298

[13] C. Feng, X. Jiang, M.-G. Jeong, H. Hong, C.-H. Fu, X. Yang, E. Wang, X. Zhu, and X. Liu, "Multitarget vital signs measurement with chest motion imaging based on MIMO radar," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 11, pp. 4735–4747, Nov. 2021.

[14] T. Stadelmayer, A. Santra, R. Weigel, and F. Lurz, "Data-driven radar processing using a parametric convolutional neural network for human activity classification," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19529–19540, Sep. 2021.

[15] M. Arsalan, A. Santra, and V. Issakov, "RadarSNN: A resource efficient gesture sensing system based on mm-wave radar," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2451–2461, Apr. 2022.

[16] M. Arsalan, M. Chmurski, A. Santra, M. El-Masry, R. Weigel, and V. Issakov, "Resource efficient gesture sensing based on FMCW radar using spiking neural networks," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2021, pp. 78–81.

[17] J. Valtl, J. Mendez, G. Mauro, A. Cabrera, and V. Issakov, "Investigation for the need of traditional data-preprocessing when applying artificial neural networks to FMCW-radar data," in *Proc. 29th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jun. 2022, pp. 1–4.

[18] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," 2017, *arXiv:1707.08945*.

[19] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against LiDAR-based autonomous driving systems," 2019, *arXiv:1907.05418*.

[20] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2267–2281, doi: 10.1145/3319535.3339815.

[21] U. Ozbulak, B. Vandersmissen, A. Jalalvand, I. Couckuyt, A. Van Messem, and W. De Neve, "Investigating the significance of adversarial attacks and their relation to interpretability for radar-based human activity recognition systems," *Comput. Vis. Image Understand.*, vol. 202, Jan. 2021, Art. no. 103111. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1077314220301338

[22] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2017, *arXiv:1712.09665*.

[23] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1924–1933.

[24] J. Valtl, J. Mendez, M. P. Cuellar, and V. Issakov, "Autonomous platform based on small-scale car for versatile data collection and algorithm verification," in *Proc. 25th IEEE Int. Conf. Pattern Recognit. (ICPR)*, Oct. 2020, pp. 1–3.

[25] J. Valtl and V. Issakov, "Frequency modulated continuous wave radar-based navigation algorithm using artificial neural network for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 567–571.

JAKOB VALTL received the B.Sc. degree in engineering science and the M.Sc. degree in electrical engineering and information technology from the Technical University of Munich, in 2016 and 2020, respectively. He is currently pursuing the Ph.D. degree with Infineon in collaboration with the Technical University of Braunschweig. He also works on artificial intelligence on edge devices in particular on adversarial attacks on radar data and secure algorithms.

VADIM ISSAKOV (Senior Member, IEEE) received the M.Sc. degree in microwave engineering from the Technical University of Munich, Munich, Germany, in 2006, and the Ph.D. degree *(summa cum laude)* from the University of Paderborn, Paderborn, Germany, in 2010. In March 2010, he joined Infineon Technologies AG, Neubiberg, Germany. Afterwards, he worked at imec, Leuven, Belgium, and Intel Corporation, before he came back to Infineon Technologies AG, in August 2015, as an mm-wave Design Lead and the Principal Engineer leading a research group working on predevelopment of millimeter-wave (mm-Wave) radar and communication products. In September 2019, he joined the University of Magdeburg, Magdeburg, Germany, as a Full Professor holding the Chair for Electronics. Since April 2021, he has been working as a Full Professor with Technische Universitat Braunschweig, Braunschweig, Germany. His research interests include mm-wave circuits, RF systems, mm-wave measurement techniques, RF-ESD, and AI techniques. He has authored or coauthored over 120 papers in international journals and conference proceedings and 11 patent applications and has published a book on *Millimeter Wave Circuits for Radar Applications*. He received an award for the outstanding dissertation from the VDI/VDE in Germany and the Best Dissertation Award from the University of Paderborn. His work has been recognized by the IEEE MTT Outstanding Young Engineer Award. He has been elected to serve as a Distinguished Microwave Lecturer of the IEEE Microwave Theory and Techniques Society (MTT-S) (2023–2025) class.