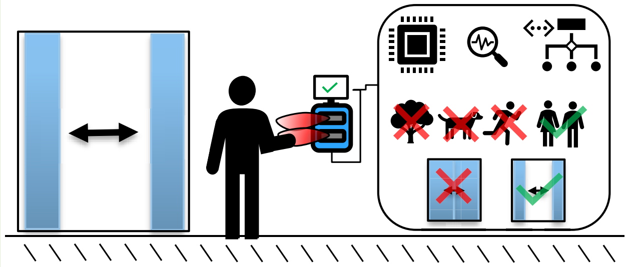
Real-Time Gesture Detection Based on Machine Learning Classification of Continuous Wave Radar Signals

Matthias G. Ehrnsperger, *Student Member, IEEE*, Thomas Brenner, Henri L. Hoese, Uwe Siart, *Member, IEEE*, and Thomas F. Eibert, *Senior Member, IEEE*



***Abstract*—Classical signal processing methodologies have been infiltrated by machine learning (ML) approaches for a long time, where the ML approaches are in particular applied when it comes to gesture recognition. In this paper, we investigate naïve gesture recognition methodologies and compare classical and novel machine learning (nML) algorithms. The considered gestures are simple human gestures such as swiping a hand or kicking with a foot. For the sake of comparability, the algorithms are assessed with respect to their true positive rate (TPR), false-positive rate (FPR), their real-time capability together with the required computational power, and their implementability on low-cost hardware. Two different data sets are utilized separately for the training process of the ML algorithms, where both have been recorded by making use of low-cost radar hardware. The results show that all ML approaches are superior to naïve gesture recognition methodologies, e.g., threshold detection. ML algorithms allow almost assured gesture detection. However, our primary contribution is a design approach for scalable neural networks (NNs) that allow such gesture recognition algorithms to be executable on low-cost microcontroller units (MCUs).**

***Index Terms*—Gesture recognition, radar, machine learning, neural networks, real-time, embedded hardware.**

I. Introduction

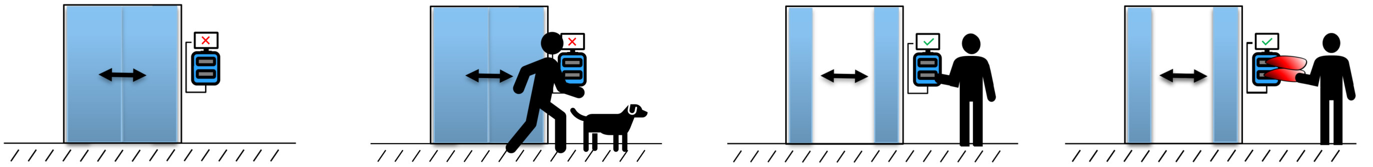
GESTURES are an essential yet mostly unrecognized part of our daily lives, most of the time they are utilized automatically and without any thinking. Every day we wave a friend good-bye, or we say no by a small finger-swing. All of these gestures and countless more are easy to recognize and interpret by humans and many of them even by animals (naturally apart from cultural and/or geographical gesture-conflicts and/or -ambiguities). The field of human gesture recognition does not only offer human-to-human or human-to-animal interaction and communication but also human-to-machine. Motion based gestures, such as a hand wave, can be recognized by machines with different sensor principles. A gesture can, for example, be detected optically with cameras [1]–[7], by employing lasers [8], [9], or by hybrid systems [10]–[12] which combine both approaches. Furthermore, it is possible to detect gestures based on radar echoes [13]–[24]. The advantage of radar systems compared to optical systems is complex: on the one hand, camera-based gesture recognition is more computationally expensive and, on the other hand, cameras are heavily dependent on weather and light conditions. Furthermore, radar systems are robust against light and weather conditions and the signals can be evaluated efficiently due to their lower resolution and are in no way inferior to cameras in terms of evaluation performance with simple gestures [25]. In general, gesture recognition by using radar signals is in the process to revolutionize the interaction of people with all kinds of electronic devices. Recently published patents, in particular by *Google LLC* [18], [20], make it irrevocably clear that in the future we might be able to control all of our computer programs, apps, games, and all imaginable systems comfortably with our hands, contactless. This can efficiently be performed with all three introduced sensor principles (camera, laser, radar), where this work is going to focus on realizing gesture recognition with radar systems. Categorizing gestures is a complex task, since gestures can be fundamentally different dependent on the

Manuscript received December 8, 2020; accepted December 11, 2020. Date of publication December 17, 2020; date of current version February 17, 2021. The associate editor coordinating the review of this article and approving it for publication was Dr. Kagan Topalli. *(Corresponding author: Matthias G. Ehrnsperger.)*

Matthias G. Ehrnsperger, Thomas Brenner, Uwe Siart, and Thomas F. Eibert are with the Chair of High-Frequency Engineering, Department of Electrical and Computer Engineering, Technical University of Munich, 80290 Munich, Germany (e-mail: m.g.ehrnsperger@tum.de).

Henri L. Hoese is with the Department of Electrical and Computer Engineering, HTWG Konstanz—University of Applied Sciences, 78462 Konstanz, Germany.

Digital Object Identifier 10.1109/JSEN.2020.3045616



(a) Automated door system with radar gesture recognition opening system. Since no gesture is conducted the door will not open.

(b) The system is not confused by the radar signatures of by-passing people or animals, so the door remains closed.

(c) Only if a correct gesture is conducted the door opens.

(d) Two radar lobes are shown in red. Each lobe represents a channel that is spatially separated and covers different areas of observation.

Fig. 1. Straightforward use-case for a radar-based gesture recognition system, the smart-opening of an automated door system [24].

person who performs the gesture. Furthermore, different kinds of gestures are different in their information content and how they are interpreted by another person/system. The gestures vary in their expressiveness and also in more mechanical matters such as the volumetric space in which the gesture is executed. Contingent upon how a gesture is conducted, the requirements for its detection change and so do the requirements on a suitable radar system. Macro gestures that are large—compared to the wavelength of the radar system—such as running, walking, or jumping are detectable with less spatial resolution than micro gestures covering a small volume, e.g., touching fingertips, or making a fist versus a flat hand. The resolution that is required for the detection process determines fundamentally the necessary bandwidth and consequently the operating frequency of the radar system. Higher frequencies tend to lead to higher cost requirements since electronic components must be manufactured more precisely. When it comes to radar hardware (HW) it is beneficial to utilize less demanding continuous-wave (CW) signals rather than the more complex frequency modulated continuous wave (FMCW) signals. An FMCW system allows to extract information about velocity and distance whereas a CW system lacks information about the distance to an object. A CW system is only able to extract the Doppler frequency shift and the corresponding intensity from a radar signal. To avoid over-engineering, and, thus, a too expensive solution, it is essential to pre-define the gestures that have to be detected. In this work, we focus on a sub-part of human activity recognition (HAR) which is the detection of natural gestures that are executed by human limbs. The gestures shall make it possible to interact with applications of our daily lives as for example the contactless opening of a door with a hand wave motion. An advanced gesture recognition door sensor only activates the door if a valid gesture is executed, see Fig. 1. An invalid signal pattern that is for instance generated by a by-passer is ignored. The valid gestures—for the sake of interactability with the low-cost radar system—have to be of sufficient volumetric size, micro gestures on the other hand, as the aforementioned "touching fingertips" are not detectable. The pre-defined gestures can be the movement signature of a human, simple hand-wave movements, or more complex gesture sequences [24]. Eventually the training dataset with its labels defines what the algorithm classifies as valid or invalid in the end. Approaching people or animals can also be defined as valid gestures, e.g., for supermarket entrances.

The analysis of the state-of-the-art for such gesture recognition tasks shows that artificial intelligence (AI) and in particular ML are employed in both science and technology where most of the publications focus on high-performance applications in combination with deep neural networks (DNNs) [26]–[28]. Here, computing power, real-time capability, and low-cost are of subordinate importance. Real-time capability means in our case that the classification result must be available within 250 ms, in order to ensure user-friendly handling of the system. This means that as soon as the user begins to execute the gesture, the algorithm starts to classify the available signals. A gesture takes approximately 1.5s, so that while the user is still executing the gesture the algorithm already presents classification probabilities whether the gesture is valid or not. Thus, there are practically no waiting times for the user. If the NN is utilised for another classification task it is perhaps necessary to vary the classification times to maintain the user-friendly handling. When it comes to Internet of Things (IoT) low-cost is more relevant, but compactness outweighs cost critical aspects so that special HW is employed that is computationally more advanced and costly than the systems that are investigated in this work [29]–[31].

Most of the high-performance applications exploit image processing methodologies, such as 2-D or 3-D convolutional neural networks (CNNs), dictionary learning, in combination with more cost-expensive radar systems [16], [32]–[39]. Such cost-expensive radars are for instance compact multi-channel radar systems with antenna-on-package (AoP) technology.

The key questions of this paper are: how can we efficiently classify simple human gestures that are recorded with a low-cost radar system? How is the mutual performance of different solutions such as threshold detection (THD) algorithms, classical machine learning (cML) methodologies, and nML methods such as NNs? Which additional functions are required by the respective approaches (THD, cML, nML) to work? Which of these three approaches allow efficient and precise gesture recognition? What accuracies are expectable and achievable with each approach? Is it possible to run the designed networks on low-cost standalone computers and low-cost MCUs? What effects occur if we want to run the gesture classification in real-time and how changes the performance of the classifiers if they are downsized during the implementation process? Which of all the investigated methods offer the overall best trade-off between performance and costs?

Binary or binomial classification categorizes elements of a dataset into one of two individual groups. The basis for this process is a so-called classification rule. Such a classification rule can be executed by hard-coded or teachable algorithms, where the latter refer to ML and are hence more related to statistical binary classification. If the number of groups in the classification process is greater than two, the process is called multiclass classification. In the following, the focus is on binary classification. The primary novelty presented in this paper is the full consideration of such a binary classification process and the transition of this classification process to low-cost HW. Here we first work on the fact that THD is not sufficient to successfully detect gestures. Then we train different cML algorithms that can solve this task much better, but cannot be implemented on inexpensive hardware. Finally, we present the actual core of this paper and what can hardly be found in the literature: a scalable NN that runs on low-cost embedded hardware and can detect gestures in real time. The advantages of using low-cost MCUs and radar HW are clear: low-cost HW is inexpensive, small, requires little power, it is easy to implement, it is available in large quantities, it is less quickly obsolete, it is can be used flexibly and often offers additional features, and it does not require a special and cost-intensive development environment. Furthermore, the presented gesture detection task can easily be mapped to many other applications where a user wants to open or activate something contactless. In Section II, the utilized radar HW is introduced. Section III presents the utilized methods. Section III-E summarizes the results and gives a discussion, before the conclusions are drawn in Section IV.

II. Radar Hardware

Two radar systems are employed for the investigations of this publication. One system operating in the X-band at 10 GHz and one system in the K-band at 24 GHz. The X-band systems has been designed with discrete components and is the result of a low-cost feasibility study, the K-band system is based on a more industrializable monolithic microwave integrated circuit (MMIC).

A. 10 GHz System

The first employed radar system operates in the X-band at 10 GHz, see Fig. 2. The system works with CW and a single mixer, hence, only the in-phase part of the signal is available. The two-channel system has an antenna constellation with monostatic channels separated by 40° in angle. Each patch antenna is placed in a polyoxymethylene (POM) housing with the corresponding radio frequency (RF) hardware. The antenna characteristic of the patch antennas is qualitatively illustrated with red lobes. The main lobe of the lower channel is vertically aligned while the main-lobe of the upper channel is tilted upwards by 40°. Both monostatic channels work with equal patch antennas [24]. The effective coverage range of the 10 GHz radar system is approximately one meter, dependent on the soil conditions. The Doppler signal in baseband is sent to an external analogue-to-digital converter (ADC), the ADC sends the data to an MCU or personal computer (PC) either via universal serial bus (USB), as illustrated, or via I^2C. The HW for this radar system costs approximately 10 EUR without manufacturing costs.

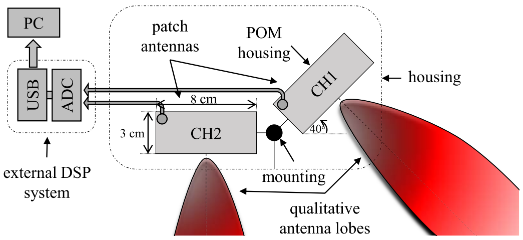


Fig. 2. Schematic of the 10 GHz CW radar gesture recording hardware.

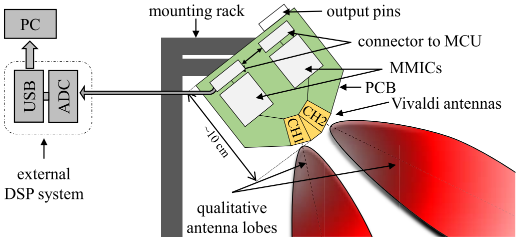


Fig. 3. The more compact 24 GHz CW radar system.

B. 24 GHz System

The second radar system is operating in the industrial, scientific and medical (ISM) band at 24 GHz, see Fig. 3. The system consists of a single printed circuit board (PCB) which contains all the required electronic components such as the power supply and the RF MMICs. These MMICs are from *Infineon's BGT24* series and set to CW with two different frequencies to avoid interference [40, p. 7]. In contrast to the 10 GHz system, the output frequencies are stable thanks to a phase locked loop (PLL). The MMICs could be set up for FMCW modulation in order to retrieve distance information. However, as the results of [23], [24] have shown, FMCW modulation is not necessarily required for excellent gesture recognition. Similar as the 10 GHz system, the 24 GHz system consists of two monostatic channels with 40° angular separation, but now with Vivaldi antennas. The antenna lobes are qualitatively illustrated in red. The main difference is that the 24 GHz system allows—thanks to the MMICs—to retrieve the in-phase part of the signal as well as the quadrature part, so that the complex signal phasor can be reconstructed. The effective distance that can be covered by the 24 GHz radar system under standard conditions (no rain, concrete floor, adult person) is approximately 50 cm. For the hardware cost, excluding manufacturing costs, just under 20 EUR are set.

III. Gesture Recognition Methodologies

A. Naïve Gesture Recognition Approach

For naïve gesture recognition, first, THD algorithms are in focus. Such algorithms can be found in all kinds of disciplines, e.g., for image segmentation in combination with evolutionary algorithms [41], [42]. Our THD algorithms are specified in a way that several threshold levels are defined, which are equally distributed for all input data and not adaptive. After the radar system has received a scattered signal from a moving object, person, or animal, it is mixed down, bandpass filtered, and down sampled. In Fig. 4, top, this process is shown for signal $S_{\mathrm{CH1}}(t)$ of channel 1. Then the Hilbert transform of $S_{\mathrm{CH1}}(t)$ is calculated to obtain the envelope $S_{\mathrm{CH1,env}} = |S_{\mathrm{CH1}}(t) + \mathrm{j}\,\mathcal{H}\{S_{\mathrm{CH1}}(t)\}|$ of the signal, see Fig. 4, middle [43]. The course of this envelope is very distinctive for a hand push gesture: in the range from about 1.2s to 1.6s the hand accelerates towards the radar system. The acceleration can be seen by the frequency in the upper plot and the amplitude of the middle plot. In the range between 1.6s to 1.8s the hand decelerates and comes to a full stop at approximately 1.8s. At the full stop, no Doppler shift can be detected, and the amplitude is almost zero as the hand lingers in front of the antenna for a brief moment. Subsequently, the hand is doing the same motion-course backwards which leads to the symmetric double peak envelope. This course of the signal is textbook style for the gesture. Based on the envelope of the signal, the threshold levels TH_1 to TH_4 have been chosen empirically by observing the dataset, see Fig. 4, bottom. The thresholds can be hard-coded or dynamic and are clocked interdependently. Now it is the task of the THD based gesture recognition algorithm to distinguish between invalid events, e.g., a by-passing person, and valid gestures that have been executed with intention. This state-of-the-art methodology does not require a training process [44], as long as the domain knowledge of the algorithm designer—that is required to set the threshold levels—is not defined as part of the training process. The theoretically achievable results of a THD algorithm in combination with Bernoulli trials are excellent [45], however, not for our dataset which is hardly linearly separable. We experimentally investigated the performance of the THD algorithm by conducting 100 valid hand kick gestures and 100 invalid cleansing gestures in front of the radar sensor. In the hand kick gesture the arm is stretched towards the radar sensor and quickly withdrawn, in the cleansing event a movement is carried out which is intended to simulate the washing of the sensor. The cleansing gesture is the most important invalid gesture throughout this work as it imitates an arbitrary and random motion. The result of this evaluation is that the multilevel THD algorithm allows only to achieve a poor TPR of 57% and an FPR of 20%. Taking all the results into consideration, it is necessary to employ more mathematically advanced methodologies from the field of cML. These methodologies require a lot of computational power, especially in the training process but promise superior results.

B. Classical Machine Learning (cML) Methodologies

As the previously presented simple THD classifier is not able to perform our classification task in a satisfactory manner the focus is shifted towards cML methodologies. The field of cML with its various methodologies is easily accessible via [46], [47] and offers high-performance algorithms to be freely trained with a dataset. For comparison, the following methods are trained and evaluated:

- naïve Bayes (NB) [46], [48],

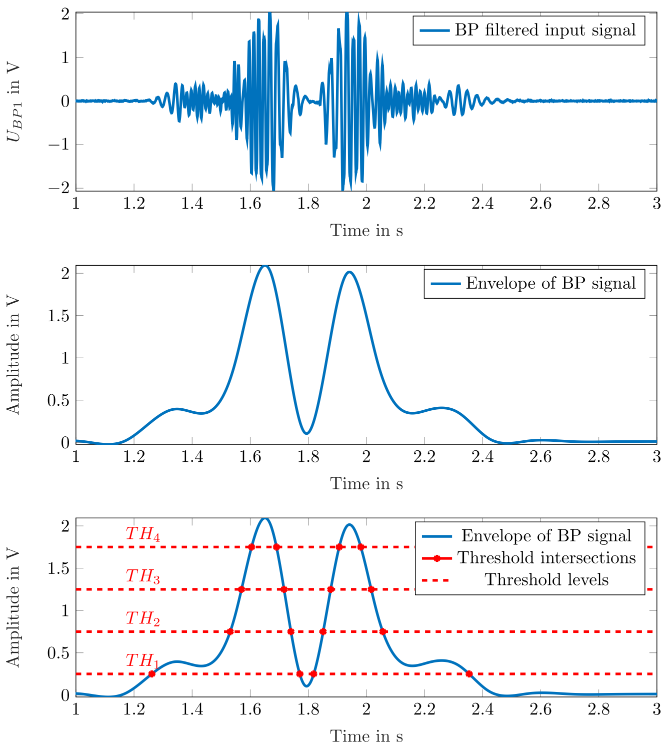


Fig. 4. Shown on top, the received signal $S_1(t)$ of the upper channel (CH1) of the 10 GHz radar system, which is received for a valid hand push event. The middle plot shows the envelope of the signal $S_{1,\mathrm{env}}$ that is extracted by utilizing the Hilbert transform. The bottom plot shows $S_{1,\mathrm{env}}$ with the corresponding threshold levels TH_1 to TH_4.

- support vector machine (SVM) [35, pp. 147],
- stochastic gradient descent (SGD) [35, pp. 119],
- decision tree (DT) [35, pp. 169].

1) Dataset: For the investigation of the cML methods, the 10 GHz CW radar system of Fig. 2 is employed. In total, we defined nine different gestures, whereby four of them are labelled as valid and five are labelled as invalid. The four valid gestures are different hand movements: straight kick, sideways kick, kick from left to right and vice versa, and a swiping movement. The five invalid gestures are a passing person, a person walking towards the sensor, a person lingering in front of the sensor, a person bending down in front of the sensor, and a scenario where the sensor is washed by a hand (this is the most challenging invalid event). The dataset consists of 3600 gestures that have been recorded individually. Every record is 5 s long and contains exactly one gesture, either valid or invalid. The duration of the gesture itself varies with the corresponding test person and the gesture, e.g., a hand push takes around 1.4s and a cleansing event can take several seconds. The cleansing event occurs when the sensor is wiped or dusted off. The data is sampled at various rates but eventually down-sampled to 1kS/s.

2) Feature Extraction: The cML algorithms require an extensive pre-processing procedure in order to create the so-called feature-vector, which is employed for both training and evaluation of the classifier. It is always necessary to construct the feature-vector, also after the classifier has been trained. The classifier can only evaluate the given signals if the features

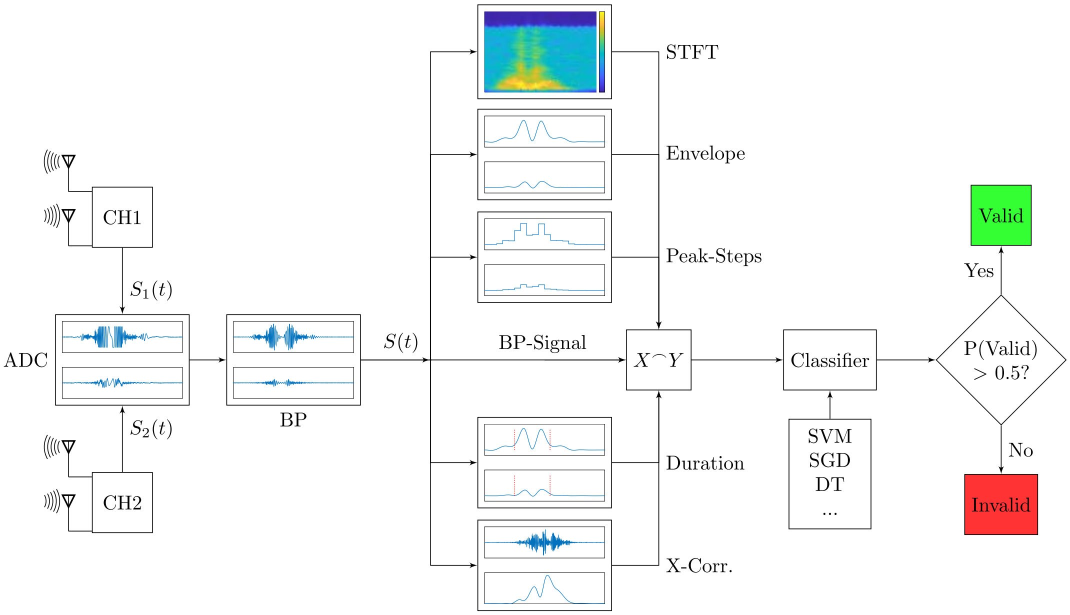


Fig. 5. Illustration of the signal flow graph for the classification process with cML methods [23].

are extracted and placed inside the feature-vector. Naturally, all information is already given in the raw time series signal, however, the cML algorithms are incapable of extracting the characteristics that are required to conduct the classification by themselves [49]. The feature extraction procedure that is employed in the following is the "hand crafted feature extraction". Hand crafted means in this context that not an algorithm decides which features are passed—and how—to the cML method but the algorithm designer. The goal of the feature extraction is to present essential characteristics of the gesture to the classifier from multiple different perspectives. For this purpose, we have decided to employ the following mathematical transforms and methods to create the feature-vector:

- bandpass (BP) filtering
- short-time-Fourier-transform (STFT)
- envelope
- stepped peak value detection
- event duration detection
- cross correlation
- envelope of cross correlation

These features were chosen empirically and are intended to create a balance between computational effort and information content. In order to finally be able to present an efficient algorithm, mathematically complex statistical feature extraction methods, i.e., the principal component analysis (PCA), were dispensed with. To examine the performance of the selected features, a sensitivity analysis was carried out, which is shown in Figure 7. In order to construct the feature-vector, first of all the raw Doppler signature is sampled at 1kS/s and subsequently BP filtered. The BP filter is hereby configured to filter out velocities below 0.45m/s and velocities above 3.00m/s. The result of the filtering process is the first part of the feature-vector itself and also the input to all the other feature extraction processes. The first feature extraction process that utilizes the BP signal as input is the STFT which constructs a spectral analysis of the input signal [50]–[52]. The next feature is the envelope which is again calculated by utilizing the Hilbert transform [43]. The envelopes of the signals are attached to the feature-vector. The next feature, the stepped peak value detection, provides a more discrete mapping. Hereby, the process utilizes a short window of 0.5s of the event and calculates the maximum to construct a histogram-like illustration. The subsequent feature is the duration of the executed gesture which is determined by employing a highly simplified THD process. The gesture duration is also attached to the feature-vector. As the radar system consists of two individual monostatic channels, all of the previously presented feature extraction processes utilize the information of both channels to construct the feature-vector, whereby all information is inserted into one feature-vector. To further exploit the information of the two channels and their connection, the cross correlation is evaluated and attached to the feature-vector. Here, again, the Hilbert transform is employed to extract the envelope of the cross correlation. In Fig. 5, the signal flow graph with all feature extraction processes and their concatenation is further illustrated [23]. The resulting feature-vector is the input of the selected classifier (i.e., SVM, SGD, DT, etc.). The classifier processes a single probability value—whether the initial signal represents a valid or an invalid gesture—based on the feature-vector. If this probability value is above a certain decision threshold (here > 0.5) the event is classified as valid, otherwise as invalid. Please note that the SVM outputs directly invalid or valid and not a probability, this slightly alters the signal flow graph [23].

3) Evaluation: All of the aforementioned classifiers are trained and evaluated in the following. Hereby, the dataset is split so that 80% of the data is presented to the cML algorithm for the training and 20% of the data for the evaluation. In order to compare the classifiers, in particular the TPR and the

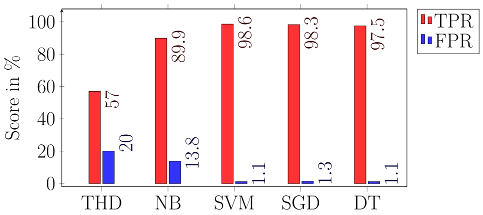


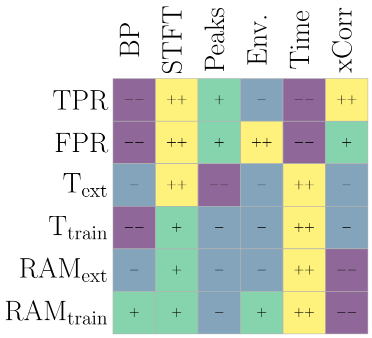
Fig. 6. Summarized results for the cML classifiers [23].

FPR are in focus. The FPR represents how frequently the door would be incorrectly activated, e.g., due to a by-passer. These two classification-performance-indicators allow to make a pre-selection, which classifier shows the best generalization towards unknown data. In combination with the learning curves that are calculated throughout this process it is possible to determine the overall performance of the algorithm. Most of the cML methodologies have parameters that allow to optimize their classification behavior.

4) Classification Results: First of all, the effect of data augmentation is investigated by employing the SVM. Three training procedures with different stages of data augmentation are evaluated, a) no augmentation, b) modest augmentation, where the original dataset is enlarged by a factor of five, and c) maximum augmentation, where the original dataset is enlarged by a factor of 45. Without augmentation, 3,670 invalid and valid events are utilized, for b) a total of 18,350 gestures are available as the data is quintupled by time-shift augmentation. For the case of c), a total of 165,150 gestures can be trained as the initial data is time-shifted and further tripled by compressing and stretching plus again tripling by amplitude up- and down-scaling. The analysis has shown that the modest augmentation of b) allows to obtain the best trade-off between increased classification performance, generalization towards new environments and test subjects, as well as the required time and computational performance for the training process. The latter requires approximately 170 h (one week) on a high-performance cluster. So, the energy that is required to train the network is in no way related to the additional classification performance. Consequently, the modest augmentation of b) is chosen.

The results of the training processes—with modest data augmentation—are illustrated in Fig. 6. Hereby, all classifiers have been optimized to increase the classification performance. The THD algorithm is added for a better overview but remains in its initial—unoptimized and unextended—state. The blue bars is the TPR (the higher the better) and the red bars are the FPR (the lower the better). The results illustrate that the THD approach is not able of classifying the gestures, the NB classifier with its little optimization potential allows to generate better results but is still of unsatisfactory performance. The more complex cML approaches—with their extensive optimization potential and parameters—allow to be tuned and boosted to obtain ideal classification results. Hereby, the SVM has not only the best TPR but also the best generalization to new data [23].

5) Feature Sensitivity Analysis: An analysis has been carried out to examine the usefulness of the selected features. In particular since state-of-the-art feature extraction methodologies, i.e. the PCA, were deliberately not employed because they are considered to be too computationally expensive. We trained the SVM six times, each time only one single feature is extracted, and evaluated the classification performance. By doing so we can can describe exactly which feature has which influence. We evaluate the TPR and FPR of each feature-wise training as classification performance indicators. Furthermore, we investigate the required time that the high-performance workstation (HPW) requires to extract the feature from the data T_{ext}, the time required to train the feature T_{train}, and additionally the random access memory (RAM) that the machine requires for the extraction and training process, respectively. The results of the feature sensitivity analysis are qualitatively illustrated in Figure 7. By training solely the BP signal the SVM cannot learn the difference between valid and invalid events and fails. The STFT of the BP signal allows to obtain a decent classification performance. All the other features, but the event duration, positively affect the classification performance. Taking the classification results and the results of the sensitivity analysis into consideration we come to the conclusion that the chosen features are reasonable. At this point it is not necessary to employ statistical feature extraction methods such as the PCA [23].



	BP	STFT	Peaks	Env.	Time	xCorr
TPR	−−	++	+	−	−−	++
FPR	−−	++	+	++	−−	+
T_{ext}	−	++	−−	−	++	−
T_{train}	−−	+	−	−	++	−
RAM_{ext}	−	+	−	−	++	−−
$\text{RAM}_{\text{train}}$	+	+	−	+	++	−−

Fig. 7. Qualitative illustration of the performances of the individual features if trained with an SVM. Here, two plus signs represent best and two minus sings the worst performance [23].

6) Implementability: The investigated cML algorithms are working well for the training and classification process—with the 10 GHz dataset—which has been completed on a HPW. The overall goal, however, is to implement a classification algorithm on embedded HW with the possibility to conduct the classification process in real-time. The SVM has been in focus due to its great generalization and performance. However, this classifier consists of more than 1000 support vectors, which requires 400 MB memory on the hard-disk. Furthermore, for the execution of the SVM, the computationally expensive feature extraction is necessary. The combination of these highly demanding requirements makes it literally impossible to implement the given SVM on embedded HW [23], [53]. Even if a less memory-demanding classifier, such as the SGD, is employed, and even if the feature extraction process is downsized to its core, it is necessary to run at least two classifiers simultaneously for real-time classification which is beyond the possibilities of low cost systems.

7) Summary and Discussion of Results: The results of the examinations of cML methodologies show that simple human gestures can be efficiently classified with a CW radar system. By detecting such human gestures resiliently, automated systems, such as sliding doors, can be operated comfortably and efficiently. The evaluation of various classifiers in combination with their optimization has shown that the binary classification process is adequately solved with cML approaches. TPRs of about 99% and FPRs of about 1% are achievable. However, the high classification performance comes at the cost of implementability. The classifiers themselves and especially the feature extraction requires a HPW to run properly.

Consequently, the classification performance of the cML methodologies should be maintained while the memory size and computational complexity has to be significantly reduced. To do so, the functionality of the cML networks has to be mapped to a NN. NNs tend to be smaller—in terms of memory size—with the same or better classification performance, this property can allow the same classification results to be achieved in real-time. Therefore, the next step is to achieve real-time gesture recognition in combination with low-cost embedded hardware.

C. Novel Machine Learning Methodologies

In the following, the investigations of nML in combination with the 24 GHz system are in focus. For the 10 GHz system also NNs have been designed, which are, however, not further addressed. An overview of the investigations with the 10 GHz system in combination with NNs can be found in [24].

1) Signal Flow Graph: The signal flow graph of the cML approach in Fig. 5 stands qualitatively for the complexity of the classification process. As the investigations have shown, highly complex cML networks are required to master the demanding task. This is different with nML methodologies and in particular NNs. No more feature extraction is necessary in the signal flow graph shown in Fig. 8. Similar to Fig. 5, in the beginning the Doppler shift of a potential gesture is detected and recorded by the monostatic radar channels CH1 and CH2. The raw signals $S_1(t)$ and $S_2(t)$ of each channel are sampled at 1kS/s and BP filtered. The resulting BP filtered signal $S(t)$, however, is neither further processed nor any features are extracted. The filtered signals $S(t)$ of both channels remain the only input signals for the selected classifier, which is based on CNN layers in our case. Here, the classifier eventually outputs the probability of whether a signal represents an invalid or a valid event, if that probability is below a certain decision threshold (here < 0.5) the event is further classified as invalid, if the probability is above the decision threshold, as valid.

2) Dataset: All recordings were carried out on concrete floor for the new 24 GHz dataset. The test subjects—and this is particularly important—represent a diverse dataset, with a 50:50%-split self-identified men and women from around 20 to 55 years old. Each test subject conducted each of the ten gestures ten times, so that 100 records of each test subject are available. With a total of 20 people, the dataset consists of 2000 supervised gestures.

Here, we defined ten different gestures, whereby again four valid and now six invalid. Increasing the number of invalid events is meaningful as the sensor is likely to be placed in crowded areas. The four valid gestures are different hand movements: kick at 0°, at 45°, and at 90°, plus a swipe movement. The six invalid gestures are the same as for the 10 GHz dataset plus a drop gesture where an object is dropped in front of the sensor.

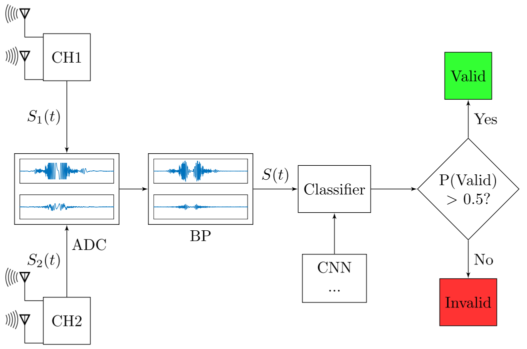


Fig. 8. Signal flow graph for classification with nML methods.

Each one of the records has a length of 3 s, is sampled at 1kS/s, consists of IQ signals, respectively, and is split 80:20% for the training- and evaluation-process. To give an overview of how different the gestures appear for the two radar systems, Fig. 9 shows the STFT for a kick-gesture for the 10 GHz as well as the 24 GHz system.

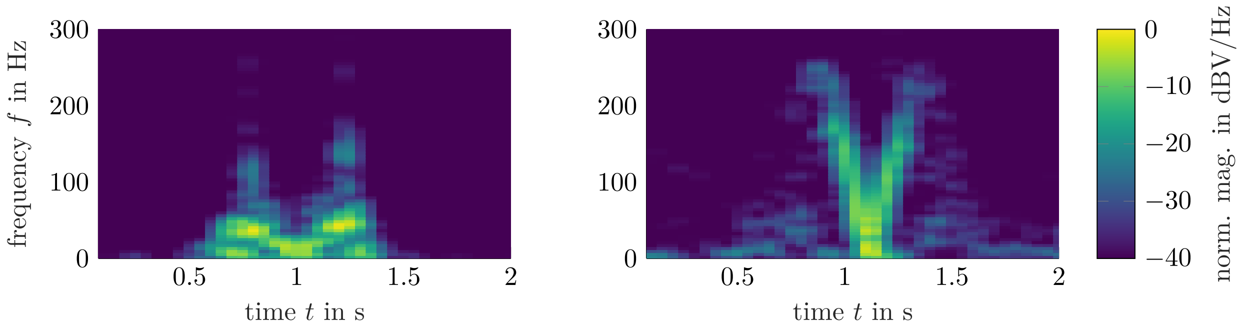
Because of the higher carrier frequency of the 24 GHz system, the Doppler frequency is higher and reaches up to approximately 250 Hz. The maximum Doppler shift of the 10 GHz system is typically below 100 Hz. Comparing the shapes of the spectrograms, the 10 GHz system has an M-like shape and the 24 GHz system a V-like shape.

Also the duration of a gesture is longer for the 10 GHz system due to the larger effective coverage range in which the radar systems can detect gestures and motions. The maximum ranges in which motions are detectable are about 1 m for the 10 GHz system and 0.5m for the 24 GHz system.

Please note that the NNs have also been evaluated with the 10 GHz dataset. For a better overview only the results of the 24 GHz dataset a presented in this paper. However, the quantitative and qualitative results are similar for both datasets and the drawn conclusion is valid.

3) Pre-Processing: For the pre-processing of each gesture, we employ only an infinite impulse response Butterworth BP. The filter has three second order sections and a defined passband from 5 Hz to 499 Hz, basically the full available spectrum of the gestures minus the mean part and most of the $1/f$-noise.

4) Data Dimensionality: Each one of the records consists of the IQ information for both channels, so that a complex phasor can be constructed. If we utilize the complex signal for an event, then we also need a more complex NN. Subsequently, a more complex NN requires more computational power. To find the best trade-off between the required dimensionality of the data and the classification performance, in the first step we designed a higher-order NN, the so-called baseline model. This baseline model is solely utilized to investigate whether IQ modulation is really required or if only the in-phase part of the signals allows to obtain satisfactory results. The baseline model is trained for six individual cases, see Table I.



(a) Spectrogram of the 10 GHz system, the event duration is about 1 s and the maximum Doppler shift about 100 Hz.

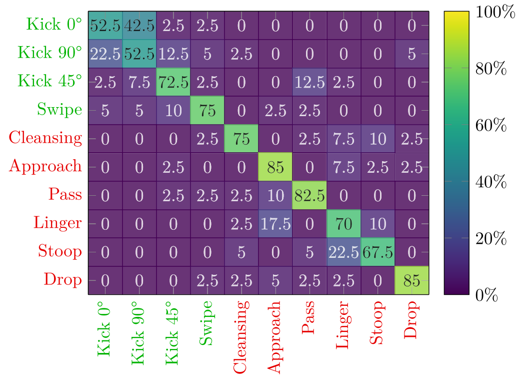
(b) Spectrogram of the 24 GHz system, the event duration is about 0.6 s and the maximum Doppler shift about 240 Hz.

Fig. 9. Spectrograms of a hand push gesture visualized for both HW prototypes, at 10 GHz and 24 GHz, respectively.

TABLE I

TPRs AND FPRs FOR THE INVESTIGATED VARIATIONS OF THE TWO CHANNEL (BOTH, UPPER, LOWER) SYSTEM, IN COMBINATION WITH IQ AND I SIGNALS, RESPECTIVELY

	both + IQ	upper + IQ	lower + IQ	both + I	upper + I	lower + I
TPR	93.1%	81.3%	87.5%	90.6%	80.6%	86.9%
FPR	2.1%	5.8%	5.0%	4.6%	9.6%	6.7%



	Kick 0°	Kick 90°	Kick 45°	Swipe	Cleansing	Approach	Pass	Linger	Stoop	Drop
Kick 0°	52.5	42.5	2.5	2.5	0	0	0	0	0	0
Kick 90°	22.5	52.5	12.5	5	2.5	0	0	0	0	5
Kick 45°	2.5	7.5	72.5	2.5	0	0	12.5	2.5	0	0
Swipe	5	5	10	75	0	2.5	2.5	0	0	0
Cleansing	0	0	0	2.5	75	0	2.5	7.5	10	2.5
Approach	0	0	2.5	0	0	85	0	7.5	2.5	2.5
Pass	0	0	2.5	2.5	2.5	10	82.5	0	0	0
Linger	0	0	0	0	2.5	17.5	0	70	10	0
Stoop	0	0	0	0	5	0	5	22.5	67.5	0
Drop	0	0	0	2.5	2.5	5	2.5	2.5	0	85

Fig. 10. Multiclass confusion matrix acquired with the baseline model. Valid gestures in green, invalid gestures in red.

With this more complex and performant baseline model, a multiclass classification between the individual gestures—exclusively for this investigation—has been conducted. All six training procedures are investigated individually. How well this baseline model is able to perform the multiclass classification is illustrated in Figure 10.

If the IQ signal of both channels is utilized, the information content is the highest and the network leads to the best classification result. However, if only the signal of the lower sensor with its I channel is utilized it is still possible to obtain a TPR of 86.9 % and an FPR of 6.7 %. In summary, the best trade-off between the required computational complexity of the input data and the classification performance, is to employ only the in-phase part of the lower sensor. However, the IQ investigation has been conducted with the deeper baseline model. This network can extract more information from the signals. Later in this section, shallower NNs are utilized which require the in-phase signals of both channels to generate satisfactory classification results.

5) Stage-Concept: For the design of the NN, a stage concept is introduced. A stage consists hereby of one convolutional layer, one pooling layer, and one dropout layer, see Fig. 11.

The number of stages that are cascaded in a row is adjustable. At the end of the N stages, the data is reshaped and eventually output through a dense layer. For a better overview the simplest constellation with one single stage is illustrated. The concept is fully scalable, whereby, with the number of consecutive stages the deepness of the network increases. If the deepness increases, so does the number of parameters and the number of multiply-accumulate (MAC) operations. This potentially leads to a more satisfactory classification process, but requires more computational power at the same time. The task of the convolutional layer is to find representative patterns in the data. The pooling operation reduces the dimensionality of the data and helps to limit the necessary computational power. To avoid overfitting, a dropout layer is utilized which cuts connections during the training process. The reshape operation at the end is required so that the data has the appropriate shape for the consecutive dense layer. As output, one single value—which represents the probability that the data is a valid event—is obtained. The activation function for the convolutional layer can be adjusted, but is the *ReLu* [54, pp. 343], [35, pp. 248] for our approach, in the dense layer a *sigmoid* [55, pp. 62], [35, pp. 136] activation function is employed.

Again, it is essential to trade-off computational complexity of the network and classification performance. Therefore, the stage concept is evaluated by re-training the network with one, two, and three consecutive stages, respectively. Where, the approach with three stages shows the best classification performance. This result is logical as most parameters are available for training. Please note that four stages are above the defined limits for the number of parameters and have not been further investigated.

6) Robustness: The designed network consists of three stages with convolutional layers and has shown great results and performance. However, the dataset consists of events that

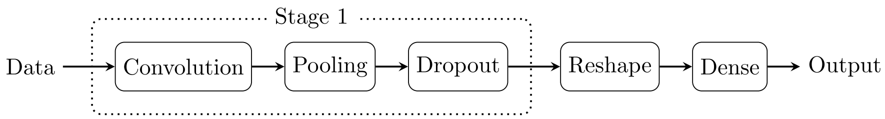


Fig. 11. Illustration of the proposed stage concept. The components of each stage are a convolutional layer of variable kernel- and filter-size, pooling (i.e., max pooling), and a dropout layer with variable dropout-probability.

have been recorded in a considerably save environment and under laboratory-like conditions. What happens if, e.g., the signal-to-noise-ratio (SNR) of the data worsens due to environmental effects [25]? For this purpose, the trained network is re-evaluated with signals from the evaluation set. However, this time these signals, which are unknown to the network, are disturbed by additive white Gaussian noise. The originally recorded data shows SNR values of more than 20 dB, so that for this investigation the SNR values are artificially set to values between 3 dB and 20 dB. With this SNR-deteriorated evaluation set, the network is tested. The results of these investigations show that by maintaining an SNR of only 5 dB the result of the classification process is not affected at all.

Additionally we conducted a cross validation with test subjects that are not part of the dataset that has been utilized for the training- and evaluation process. Three test subjects conducted the aforementioned gestures whereby the presented classification performances could be maintained.

7) Real-Time Classification: The presented network shows a great classification performance even under deteriorated SNR conditions. However, the network still utilizes a center-of-gravity cut-out of the 3 s gesture-records as input. This is too slow and not real-time, so the process has to be fundamentally changed. As mentioned in the introduction, the term *real-time* is representative of the fact that a first classification result is available within 250 ms.

Regarding this fact, the NN is trained with a short window length of 1 s. This has the advantage that the NN requires only 1000 datapoints to make a prediction, whether the corresponding 1 s-snippet is part of a valid- or of an invalid-event. The investigated real-time approach uses a sliding window technique with a window length of 1 s, see Fig. 12. The window is shifted by 250 ms for every prediction. Time-shift leads to overlapping windows which guarantees that every data sample is part of the prediction. It is mandatory that the time duration for preprocessing and model execution is shorter than the time-shift. When the application is started the data acquisition begins. After 1 s, when 1000 datapoints are collected and fed into the NN, the first classification result can be obtained. Then after every 250 ms another classification result is obtainable. Therefore, the approach has a short settling time at the beginning.

8) Real-Time Optimization: The real-time approach is inevitable. However, the shorter observation windows of the real-time approach reduce the available information, so that the FPR increases and the overall classification performance reduces consequently. However, with the introduced real-time approach of Fig. 12, there is a new possibility for optimization. Illustrated are the two radar channels (CH1 and CH2) that are mixed down to direct current (DC) and sampled at 1kS/s by the ADC. Subsequently the signals are BP filtered with the presented infinite impulse response (IIR) Butterworth filter. In the following, for a better overview, only the signal of CH1 is illustrated, the signals of CH2 remain part of the signal flow. The ADC continuously samples the input and saves the signal in a ring-buffer, the overall result of the sampling process is illustrated as a 2 s long signal $S(t)$. Throughout, the real-time process, snippets of the ring buffer are loaded as slices (Slice 1-5). Each one of these slices is individually given to the NN classifier. The NNs output a probability between 0 and 1 which corresponds to one of the slices. If the output probability is above the decision threshold, here illustrated as 0.5, it is set to a logical 1, if the probability is below the decision threshold, it is set to a logical 0. Hereby, 1 stands for a valid event and 0 for an invalid event. The continuous classification process allows to obtain a vector of binary classification results, as illustrated as red-green vertical stack. For the final evaluation of the gesture, it is possible to utilize the result of a single 1 or to combine several positive classification results with an additional logic. The decision as to whether a gesture is ultimately defined as valid or invalid is fundamentally dependent on the output value of the NN, but also on the defined threshold. The threshold, which is 0.5 by default, can be optimized using the so-called receiver operating characteristic (ROC) curve. The real-time approach updates the classification result each 250 ms, so that for a kick gesture—that takes approximately 1.5s—at least six classification results are available. These classification results, which are logical ones and zeros, are collected in the aforementioned vector. This vector—of binary classification results—can be exploited, so that the final classification result is only valid, if, e.g., two or more consecutive ones are inside the vector. We can take advantage of this fact. A simple logic, which is activated at the output, observes successive windows and decides on the basis of their sequence, whether a gesture is ultimately defined as valid or invalid. As a result, the observation period is extended accordingly, and the output of the result is delayed. As a trade-off between performance gain and latency, two successive windows are analyzed. This so-called logic optimization allows the TPR to be increased by 3.8 %. However, one parameter remains unoptimized: the threshold itself. By exploiting the information of the ROC curve, where FPR and TPR are traded-off against each other, the best balance has to be found. In order to expand the information from the ROC curve, all output values of the NN have been analyzed in a histogram. The result is that the classification results are mostly clear, i.e., almost 0 for invalid and almost 1 for valid. If we take this information and the information from the ROC curve together, we can draw a conclusion about how to optimize the threshold: increase significantly,

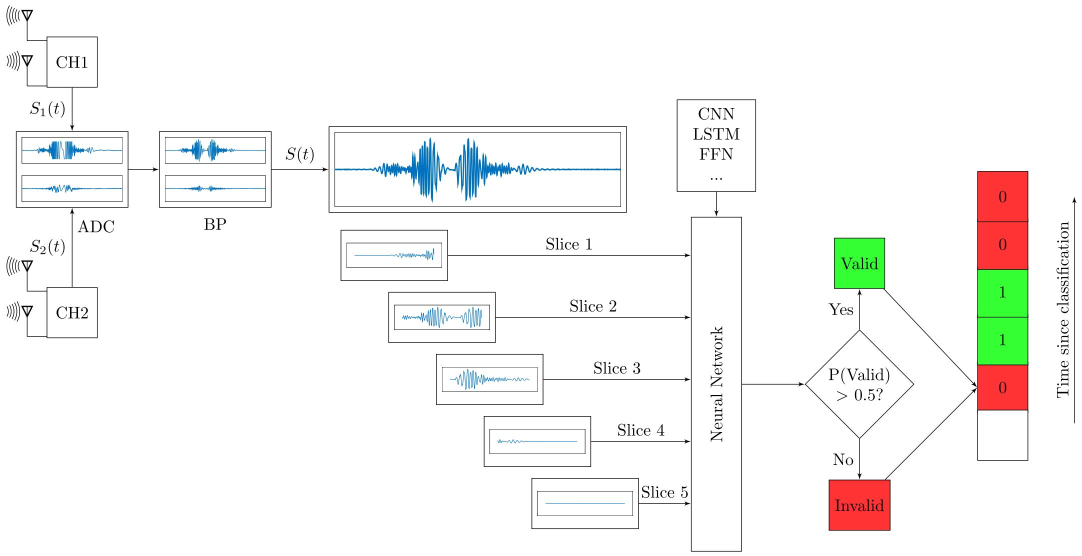


Fig. 12. Signal flow graph of the real-time approach for nML classification. The gesture is divided into slices that are presented to the NN.

TABLE II
TECHNICAL SPECIFICATIONS OF THE TWO SELECTED HW PLATFORMS. *1 KiB $\widehat{=}$ 2^{10} Byte

Platform	Raspberry Pi 3 B+	STM32F303RE Nucleo
Processor	4 x ARM Cortex-A53	ARM Cortex-M4
RAM	1024 MB	80 KB SRAM
ROM	max. 64 GB	512 KB Flash
Clock	1.2 GHz	72 MHz
RAM	45.0 KiB*	6.2 KiB
Price	30.67 EUR [58]	3.48 USD (quant.: 10k) [59]

TABLE III
OPTIMIZATION AND DOWNSIZING PROCESS OF THE NNS, THE INITIAL NETWORK CONSISTED OF THREE STAGES WITH A TOTAL OF 13,281 PARAMETERS THAT TOOK THE STM32 NUCLEO BOARD ABOUT 1425 ms TO CLASSIFY ONE SINGLE GESTURE

	Initial network (CNN_1)	Optimized network (CNN_4)
Stages	3 @ 10x50	2 @ 3x(50/25)
Parameters	13, 281	1, 582
MAC	5, 786, 290	722, 266
t_c	1425.2 ms	216.0 ms
ROM	51.9 KiB	6.2 KiB
RAM	45.0 KiB	24.4 KiB

ideally to a value of about 0.9. Accordingly, the threshold is increased to over 0.9 in the next step. Now, there is an improvement in TPR of 1.9 %, but unfortunately the FPR also increases significantly. This is an unacceptable outcome so that the optimization is redesigned. The logic—which outputs valid for two consecutive ones in the classification vector—can only be of limited help and is omitted in the following. The key ultimately lies in the ROC curve and the decision threshold. This decision threshold is now, again taking into account the classification histogram and the ROC curve, set to a value above 0.9, without logic. As a result, the TPR increases eventually from 91% to 94.3 % and the FPR reduces from 12.1 % to 2.5 %.

9) Real-Time Embedded Classification: The designed NN is now able to classify gestures in real-time, so that a classification result is available at least every 250 ms. It is also able to do so, not only on a HPW, but also on embedded HW such as the *Raspberry Pi 3B+* [56]. This Raspberry Pi 3B+, however, is more a standalone computer as it is a low-cost embedded hardware. When it comes to low-cost, then the *STM32-Nucleo* board [57] is a better choice. It is a classical MCU, which is beyond the computational performance of the Raspberry Pi. The technical specifications of both are summarized in Table II.

The STM32 Nucleo low-cost platform has reduced memory storage (RAM and read only memory (ROM)) and offers less computational power. In particular the clock frequency of 72 MHz is fairly low compared to the Raspberry Pi with 1.2GHz. The real-time model, however, still being unoptimized for this purpose, is able to run on the Nucleo board after an extensive compilation process. The classification of one gesture unfortunately takes 1425 ms, which is by far too long. With the presented stage concept, first of all the number of stages can be modified. Fewer stages mean fewer parameters. Furthermore, the size of each kernel and the number of filters can be configured. Reducing all these numbers also affects the classification performance. In a multi-step optimization procedure, the NN has been gradually downsized, retrained, evaluated, compiled, and timed. This procedure was repeated several times until the optimization minimum was reached. With this optimization the total number of parameters of the NN was reduced from 13,281 to 1,582 and consequently the duration of the classification process on the STM32 Nucleo board from 1425 ms to 216 ms, see Table III. In this process, the number of MAC operations have been reduced by 87.5 % while maintaining a TPR of 93.0 % and an FPR of 1.25 %.

D. Reducing the Computational Load

As presented in [23], it is possible to reduce the computational load that is generated by the classifiers, especially for the cML classifiers. The computational load is in particular reduced, if only one of the two radar channels is utilized and if the feature-vector is downsized to the essential features. All of which reduces the required computations and is beneficial for the implementation of embedded hardware. However, the cML methodologies remain too hungry for computing power so that an implementation of a low-cost MCU such as the STM32 Nucleo is still not possible. Reducing the input to only one radar channel is also possible for the approach with nML NNs as presented in [24]. However, if only one channel is utilized, the real-time approach is not able to obtain satisfactory classification results. The real-time approach—where the information density for the classifier is significantly lower—requires the information of both channels to generate a satisfactory classification performance. Naturally, a deeper and more complex network, i.e., a deep convolutional neural network (DCNN), would allow to maintain an acceptable classification performance but comes at the cost of implementability on the presented low-cost hardware. The best trade-off—for the less information dense real-time classification process—is to utilize both channels. If more powerful low-cost embedded HW is available in the future, it is likely that only one channel is required to maintain the results.

E. Discussion

Our investigations have shown that the naïve THD approach is incapable to correctly classify the gestures. However, cML approaches, i.e., the SVM, on the other hand allow to obtain TPRs of 99% and FPRs of 1%. To obtain these results a computationally expensive feature extraction process is required and even more computational power is needed to classify the feature-vector with the corresponding network. As a result, the cML approaches remain insufficient in their price-performance. The approach with nML methods and in particular with CNNs allows to create networks that classify in real-time on low-cost embedded hardware. The presented stage-concept allows to scale the networks from high TPR mutliclass classifying NNs to low-power real-time classifying NNs. Table IV further summarizes the investigations of cML versus nML. The class of cML methods is hereby represented by the SVM as it performed best upon the cML methods in our investigations. For the NNs, four final networks CNN_{1-4} were presented. Each offers different properties in combination with different classifications performances. CNN_{1+2} are for offline classification only. CNN_3 classifies in real-time but only on more expensive HW such as the Raspberry Pi 3B+. The optimized version CNN_4 classifies in real-time on the low-cost embedded platform STM32 Nucleo. All CNNs allow to achieve TPRs of 93.0 % and higher and FPRs down to 1.3 %.

TABLE IV
OVERVIEW OF THE OBTAINABLE RESULTS WITH DIFFERENT APPROACHES

Method	TPR	FPR	Offline	Multiclass	Real-time (RT)	RT embedded
cML	98.6%	1.1%	X			
CNN_1	97.3%	1.7%	X			
CNN_2	96.0%	2.8%		X		
CNN_3	94.3%	2.5%			X	
CNN_4	93.0%	1.3%				X

Regarding the comparability of the results to the presented literature of the introduction. There, many high-performance methodologies are investigated to achieve the best possible performance, i.e., by employing 3D CNN classifiers with triplet loss and DCNNs [26]–[28]. However, our presented low-cost approach is computationally inexpensive and cannot be meaningfully benchmarked with these high-performance approaches. Furthermore, the presented STM32 MCU has limited memory so that such high dimensionality classification networks are too large in memory to be implemented and could not be made runnable in an acceptable time-frame.

IV. Conclusion

The goal of this work was to investigate how simple human gestures—that have been recorded with low-cost CW radar systems—can be classified. With the classification of these gestures, it shall be made possible to operate everyday applications, such as an automated door, comfortably. We presented two radar prototypes, one operating at 10 GHz and one operating at 24 GHz. For each of the two radar systems a comprehensive dataset has been generated with the help of various test subjects. To conduct the binary classification task, firstly a THD approach has been presented and evaluated. The results for the THD approach have shown that the complexity of the gestures cannot be reflected by the THD classifier, which means that the classifier fails during the classification process. Subsequently, we presented the implementation and evaluation of various cML classifiers. Our investigations have shown that these more complex approaches are able to binarily classify the recorded gestures. It is possible to achieve TPRs of about 99% and FPRs of about 1%. The cML classification process is resilient and especially the SVM achieves great generalization towards new and unknown data. However, the cML algorithms have one major downside: their implementability. It is hardly possible to implement the algorithms on low-cost embedded HW and the classification process takes too long. Therefore, in the next step we investigated the possibilities of nML methodologies such as NNs and in particular of CNNs. We proposed a stage-wise scalable NN design guide that allows to create multiclass-classification capable NNs, real-time capable NNs, as well as downsized NNs that are suitable for low-cost embedded hardware. All three design concepts individually trade-off classification performance versus costs.

References

[1] K. Ishii, S. Zhao, M. Inami, T. Igarashi, and M. Imai, "Designing laser gesture interface for robot control," in *Human-Computer Interaction*, T. Gross *et al.*, Eds. Berlin, Germany: Springer, 2009, pp. 479–492.

[2] P. Breuer, C. Eckes, and S. Müller, "Hand gesture recognition with a novel IR time-of-flight range camera—A pilot study," in *Computer Vision/Computer Graphics Collaboration Techniques*. Berlin, Germany: Springer, 2007, pp. 247–260.

[3] W. Du and H. Li, "Vision based gesture recognition system with single camera," in *Proc. 5th IEEE Int. Conf. Signal Process.*, Aug. 2000, pp. 1351–1357.

[4] B. Ionescu *et al.*, "Using a NIR camera for car gesture control," *IEEE Latin Amer. Trans.*, vol. 12, no. 3, pp. 520–523, May 2014.

[5] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016.

[6] S. Lian, W. Hu, and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 107–115, Feb. 2014.

[7] J. P. Sahoo, S. Ari, and D. K. Ghosh, "Hand gesture recognition using DWT and F-ratio based feature descriptor," *IET Image Process.*, vol. 12, no. 10, pp. 1780–1787, Oct. 2018.

[8] S. Perrin, A. Cassinelli, and M. Ishikawa, "Gesture recognition using laser-based tracking system," in *Proc. 6th IEEE Int. Conf. Automat. Face Gesture Recognit.*, May 2004, pp. 541–546.

[9] P. R. Bhamre, R. G. Chauhan, and S. Kumar, "Gesture recognition using laser sensor enhanced with different parameters," in *Proc. IEEE Global Conf. Wireless Comput. Netw. (GCWCN)*, Dec. 2014, pp. 79–83.

[10] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659–1673, Sep. 2016.

[11] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[12] H. M. Jais, Z. R. Mahayuddin, and H. Arshad, "A review on gesture recognition using kinect," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Aug. 2015, pp. 594–599.

[13] M. Ritchie, A. Jones, J. Brown, and H. D. Griffiths, "Hand gesture classification using 24 GHz FMCW dual polarised radar," in *Proc. Int. Conf. Radar Syst. (Radar)*, 2017, pp. 1–6.

[14] M. G. Amin, Z. Zeng, and T. Shan, "Hand gesture recognition based on radar micro-Doppler signature envelopes," 2018, *arXiv:1811.12467*. [Online]. Available: http://arxiv.org/abs/1811.12467

[15] T. Sakamoto, X. Gao, E. Yavari, A. Rahman, O. Boric-Lubecke, and V. M. Lubecke, "Hand gesture recognition using a radar echo I–Q plot and a convolutional neural network," *IEEE Sensors Lett.*, vol. 2, no. 3, pp. 1–4, Sep. 2018.

[16] J.-W. Choi, S.-J. Ryu, and J.-H. Kim, "Short-range radar based real-time hand gesture recognition using LSTM encoder," *IEEE Access*, vol. 7, pp. 33610–33618, 2019.

[17] H.-S. Yeo and A. Quigley, "Radar sensing in human-computer interaction," *Interactions*, vol. 25, no. 1, pp. 70–73, Dec. 2017.

[18] P. M. Amihood and I. Poupyrev, "Advanced gaming and virtual reality control using radar," U.S. Patent 2019 0232156 A1, Aug. 1, 2019.

[19] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sens.*, vol. 11, no. 9, p. 1068, May 2019.

[20] P. M. Amihood and I. Poupyrev, "Advanced gaming and virtual reality control using radar," U.S. Patent 10300370 B1, May 28, 2019.

[21] I. Poupyrev, "Radar-based gesture recognition," German Patent 112015003655 T5, May 11, 2017.

[22] I. Poupyrev, "Radar-based gesture recognition," WO Patent 2016022764 A1, Feb. 11, 2016.

[23] M. G. Ehrnsperger, H. L. Hoese, U. Siart, and T. F. Eibert, "Performance investigation of machine learning algorithms for simple human gesture recognition employing an ultra low cost radar system," in *Proc. Kleinheubacher Tagung*, Miltenberg, Germany, Sep. 2019, pp. 1–4.

[24] M. G. Ehrnsperger, T. Brenner, U. Siart, and T. F. Eibert, "Real-time gesture recognition with shallow convolutional neural networks employing an ultra low cost radar system," in *Proc. German Microw. Conf.*, Cottbus, Germany, Mar. 2020, pp. 88–91.

[25] M. G. Ehrnsperger, U. Siart, M. Moosbühler, E. Daporta, and T. F. Eibert, "Signal degradation through sediments on safety-critical radar sensors," *Adv. Radio Sci.*, vol. 17, pp. 91–100, Sep. 2019.

[26] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proc. 29th Annu. Symp. User Interface Softw. Technol. (UIST)*. Tokyo, Japan: Association for Computing Machinery, 2016, pp. 851–860.

[27] S. Hazra and A. Santra, "Short-range radar-based gesture recognition system using 3D CNN with triplet loss," *IEEE Access*, vol. 7, pp. 125623–125633, 2019.

[28] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3041–3048, Apr. 2019.

[29] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, "TinyRadarNN: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars," 2020, *arXiv:2006.16281*. [Online]. Available: https://arxiv.org/abs/2006.16281

[30] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64-mW DNN-based visual navigation engine for autonomous nano-drones," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8357–8371, Oct. 2019.

[31] M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti, and L. Benini, "Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 244–256, Apr. 2020.

[32] J. Kwon and N. Kwak, "Human detection by neural networks using a low-cost short-range Doppler radar sensor," in *Proc. IEEE Radar Conf.*, May 2017, pp. 0755–0760.

[33] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[34] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 1–7.

[35] A. Géron, *Hands-On Machine Learning With Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, N. Tache, Ed. Sebastopol, CA, USA: O'Reilly Media, 2017.

[36] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "Centralized class specific dictionary learning for wearable sensors based physical activity recognition," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[37] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "Maximum correntropy based dictionary learning framework for physical activity recognition using wearable sensors," in *Advances in Visual Computing*, G. Bebis *et al.*, Eds. Cham, Switzerland: Springer, 2016, pp. 123–132.

[38] S. M. Mathews, "Dictionary and deep learning algorithms with applications to remote health monitoring systems," Ph.D. dissertation, Univ. Delaware, Newark, DE, USA, 2017.

[39] S. M. Mathews, "Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review," in *Proc. Comput. Conf. Intell. Comput.*, Jul. 2019, pp. 1269–1292.

[40] *Datasheet BGT24 ATR11—Silicon Germanium 24 GHz Transceiver MMIC*, IT AG, Munich, Germany, Sep. 2013.

[41] D. Oliva, S. Nag, M. A. Elaziz, U. Sarkar, and S. Hinojosa, "Multilevel thresholding by fuzzy type II sets using evolutionary algorithms," *Swarm Evol. Comput.*, vol. 51, Dec. 2019, Art. no. 100591.

[42] M. A. Elaziz and S. Lu, "Many-objectives multilevel thresholding image segmentation using knee evolutionary algorithm," *Expert Syst. Appl.*, vol. 125, pp. 305–316, Jul. 2019.

[43] S. Marple, "Computing the discrete-time 'analytic' signal via FFT," in *Proc. 31st Asilomar Conf. Signals, Syst. Comput.*, 1997, pp. 1322–1325.

[44] P. Gubrele, R. Prasad, P. Saurabh, and B. Verma, "Advance morphological filtering, correlation and convolution method for gesture recognition," in *Proc. 7th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Nagpur, India, Nov. 2017, pp. 153–157.

[45] J. Kreimer and M. Dror, "The monotonicity of the threshold detection probability in a stochastic accumulation process," *Comput. Oper. Res.*, vol. 17, no. 1, pp. 63–71, Jan. 1990.

[46] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[47] L. Buitinck *et al.*, "API design for machine learning software: Experiences from the scikit-learn project," in *Proc. ECML PKDD Workshop, Lang. Data Mining Mach. Learn.*, 2013, pp. 108–122.

[48] H. Zhang. (2004). *The Optimality of Naive Bayes*. Accessed: Mar. 18, 2019. [Online]. Available: https://www.cs.unb.ca/hzhang/publications/FLAIRS04ZhangH.pdf

[49] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[50] S. Abdulatif, Q. Wei, F. Aziz, B. Kleiner, and U. Schneider, "Micro-Doppler based human-robot classification using ensemble and deep learning approaches," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2018, pp. 1043–1048.

[51] Z. Liu, L. Li, H. Xu, and H. Li, "A method for recognition and classification for hybrid signals based on deep convolutional neural network," in *Proc. Int. Conf. Electron. Technol. (ICET)*, May 2018, pp. 325–330.

[52] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 8–12, Jan. 2016.

[53] D. Yang, N. Chhatre, F. Campi, and C. Menon, "Feasibility of support vector machine gesture classification on a wearable embedded device," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, Vancouver, BC, Canada, May 2016, pp. 1–4.

[54] C. Sammut, *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. New York, NY, USA: Springer, 2017.

[55] N. J. Nilsson, *Introduction to Machine Learning*. Stanford, CA, USA: Stanford Univ., Robotics Laboratory, Department of Computer Science, 1996.

[56] (2019). *The Raspberry Pi Foundation Raspberry Pi 3Model B+*. Accessed: Apr. 27, 2020. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf

[57] *UM1724 User Manual—STM32 Nucleo-64 Boards (MB1136)*, STMicroelectronic, Geneva, Switzerland, Apr. 2019.

[58] *Raspberry Pi 3+. Reichelt*. Accessed: Mar. 29, 2020. [Online]. Available: https://www.reichelt.de/raspberry-pi-3-b-4x-1-4-ghz-1-gb-ram-wlan-bt-raspberry-pi-3b-p217696.html

[59] STMicroelectronic. *STM32F303RE*. Accessed: Mar. 29, 2020. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f303re.html

Matthias G. Ehrnsperger (Student Member, IEEE) received the B.Eng. degree in electrical engineering and information technology from the Technical University of Applied Sciences Regensburg, Regensburg, Germany, in 2015, and the M.Sc. degree in electrical engineering and information technology from the Technical University of Munich, Munich, Germany, in 2017.

From 2017 to 2020, he has been a Research Assistant with the Chair of High-Frequency Engineering, Department of Electrical and Computer Engineering, Technical University of Munich. In 2020, he became an Analog Hardware Development Engineer for spectrum analyzers at Rohde & Schwarz, Munich. His current research interests include microwave circuits, radar systems, rapid prototyping methodologies, as well as statistical- and machine learning-based signal processing.

Thomas Brenner, photograph and biography not available at the time of publication.

Henri L. Hoese graduated from the HTWG Konstanz—University of Applied Sciences in 2013. He formed knowledge in the field of electrical engineering and software information systems. During his bachelor's degree, he completed an internship at a big automotive supplier in the field of research and development. In his studies, he mainly focused on software development. After his bachelor's degree, he also continued to study electrical systems in Konstanz. His main focuses include software development, embedded sensor techniques, and artificial intelligence. He wrote his master thesis in corporation with an automotive supplier. His task was the investigation of various AI algorithms to classify human gestures based on radar sensor signals. He is currently applying his knowledge in the field of software development at a Swiss OEM for public transport.

Uwe Siart (Member, IEEE) received the Dipl.-Ing. degree from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1996, and the Dr.-Ing. degree from the Technical University of Munich, Munich, Germany, in 2005. Since 1996, he has been with the Chair of High-Frequency Engineering, Department of Electrical and Computer Engineering, Technical University of Munich, where he became a Senior Research Associate in 2005. His research interests include the fields of signal processing and model-based parameter estimation for millimeter-wave radar signal processing and high-frequency measurements. He is working on stochastic electromagnetic wave propagation, remote sensing the atmosphere, low-power radar sensors, microwave reflectometry, and passive millimeter-wave components.

Thomas F. Eibert (Senior Member, IEEE) received the Dipl.-Ing. (FH) degree in electrical engineering from Fachhochschule Nürnberg, Nuremberg, Germany, in 1989, the Dipl.-Ing. degree in electrical engineering from Ruhr-Universität Bochum, Bochum, Germany, in 1992, and the Dr.-Ing. degree in electrical engineering from Bergische Universität Wuppertal, Wuppertal, Germany, in 1997.

He is currently a Full Professor of High-Frequency Engineering with the Technical University of Munich, Munich, Germany.