NLOS Detection using UWB Channel Impulse Responses and Convolutional Neural Networks

Maximilian Stahlke* stahlkema64636@th-nuernberg.de
Sebastian Kram†‡ sebastian.kram@iis.fhg.de
Christopher Mutschler† christopher.mutschler@iis.fhg.de
Thomas Mahr* thomas.mahr@th-nuernberg.de

*Nuremberg Institute of Technology Georg Simon Ohm

†Precise Localization and Analytics Department Fraunhofer Institute for Integrated Circuits IIS

‡Institute of Information Technology (Communication Electronics), Friedrich-Alexander University (FAU) Erlangen-Nürnberg

***Abstract*—Indoor environments often pose challenges to RF-based positioning systems. Typically, objects within the environment influence the signal propagation due to absorption, reflection, and scattering effects. This results in errors in the estimation of the time or arrival (TOA) and hence leads to errors in the position estimation. Recently, different approaches based on classical, feature-based machine learning (ML) have successfully detected such obstructions based on CIRs of ultra wideband (UWB) positioning systems.**

This paper applies different convolutional neural network architectures (ResNet, Encoder, FCN) to detect non line-of-sight (NLOS) channel conditions directly from the CIR raw data. A realistic measurement campaign is used to train and evaluate the algorithms. The proposed methods highly outperform the feature-based ML baselines while still using low network complexities. We also show that the models generalize well to unknown receivers and environments and that positioning filters benefit significantly from the identification of NLOS measurements.

***Index Terms*—NLOS detection, CNN, CIR, UWB, IIOT.**

I. Introduction

Many technologies have been proposed for indoor RF-based positioning like Ultra-wideband (UWB) [1], WiFi [2] and RFID [3]. UWB systems, which make use of the time-of-flight (ToF) estimate the time-of-arrival (TOA) or time-difference of arrival (TDOA), achieve accuracies in the decimeter range, given mild propagation conditions.

However, many indoor environments such as industrial production areas often include interfering objects that introduce effects such as absorption, diffraction, and scattering. In many cases, the line-of-sight (LOS) between transmitter and receiver is completely obstructed. This causes a drastic deterioration in positioning accuracy for many classical approaches, which use, for instance, least squares estimators and Bayesian tracking filters to estimate the positions from ToA measurements. Hence, a reliable detection of channel conditions, especially the non-line-of-sight (NLOS) condition, is an enabler for precise and robust localization in such harsh environments.

This paper proposes an NLOS detection approach that uses convolutional neural networks (CNNs) directly on the channel impulse responses (CIRs) from UWB-based radio systems. We conduct a realistic and extensive NLOS detection study that uses two different environments and six receivers. We train three different network architectures (i.e., Encoder [4], ResNet [5] and FCN [5]) with various sizes (resulting in a total of 20 different models) and evaluate their performance based on their classification accuracy and their ability to improve a Bayesian tracking filter. Our experiments prove the efficiency of CNN-based approaches for the NLOS detection tasks on CIRs and analyze their performance in consideration to the available capacity of the networks.

The remainder of the paper is structured as follows. Sec. II reviews related work. We provide details about the classifiers in Sec. III. Sec. IV describes the experimental setup, followed by the evaluation of the NLOS detection approach in Sec. V.

II. Related work

Several techniques have been proposed to detect NLOS using UWB CIRs. Most of them extract features from the CIR and then classify different channel conditions. Kolakowski et al. [6] use the power of the first path by applying thresholds for the different channel states, assuming that there are varying attenuations on the signals. Ismail et al. [7] proposed a statistical approach that employs likelihood-ratio tests of features from the CIR to detect NLOS. However, such approaches either have limited detection rates or require a lot of manual tuning.

In the recent years, machine learning (ML) approaches have been applied for NLOS detection. Support vector machines (SVMs) are often used as they are adaptable, efficient and accurate enough for various classification problems. Wu et al. [8] use an SVM on time-difference-of-arrival (TDOA) measurements of multiple base stations to detect NLOS of the receiver. Li et al. [9] proposed a feature-based classification approach using a least-squares SVM, i.e., a non-parametric form of SVM, to detect NLOS. There also exist approaches that use SVMs on discretized CIRs instead of extracted features [10]. Apart from SVMs also feature-based decision trees have been used [11]. All the above mentioned approaches use features extracted from the CIR to train a classifier that detects the NLOS. While such approaches perform well in

practice, their feature extraction does not consider the CIR's full information.

Recently deep learning (DL) has successfully been applied to many related tasks such as velocity estimation [12] and positioning with CIRs [13], [14]. DL uses raw data to extract the features automatically rendering a manual feature extraction unnecessary [15]. Such methods have been successfully applied to improve the TOA estimation in UWB systems [16] and to classify channels conditions [17]. Cwalina et al. [18] use a Deep Feedforward Neural Network (DFNN) that uses the total power and the power of the first component of the CIR. Liao et al. [19] proposes a downlink channel estimation network employing CNNs for feature extraction and recurrent networks for channel estimation. However, those methods did not explicitly focus on the NLOS detection.

Others identify NLOS using channels state information and received signal strength indicators of WLAN devices [20], in GNSS receivers [21], or by employing the short-time Fourier transform of the CIRs [22]. Abadi et al. [23] show the feasibility of NLOS detection using CNN models and simulated CIRs with different signal to noise ratios. Bregar et al. [24] apply a singular CNN architecture for image classification adapted to one dimensional time series data, which exploits the raw CIR to detect obstruction of the signal. In contrast to them we evaluate different CNN architectures designed for time series classification and focus on the influence of the model complexity on the classification performance.

III. Classification Algorithms

A. Support vector machine

As a baseline we used an SVM with two different kernels, i.e., a linear and a radial kernel. These kernels have previously been successfully applied for NLOS detection [9], [10], [25]. Based on literature [26], we choose appropriate features for our SVMs: energy index, correlation maximum, energy decay time index, peak decay exponent and time-frequency domain features (bandwidth, centroid, rolloff, flatness). However, the information content of the spectral features is widely redundant, therefore a dimensionality reduction using a principal component analysis is applied, resulting in two features. In total, 6 features are used for classification with our SVMs.

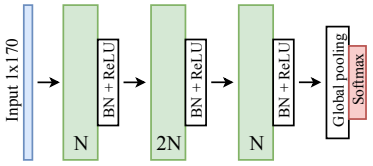
B. Convolutional neural networks

Classic feed-forward artificial neural networks (FFNN) consist of many interconnected neurons, arranged in fully connected layers. Neurons consist of several input connections and an output function. A weight is assigned to every input connection, which controls the influence of the preceding neuron on the output function. Non-linear transformations, like the sigmoid function or the rectified linear unit (ReLU) [27], are used as output functions. In the case of a feed-forward neural network, only connections from the input layers to the succeeding layers are possible. A neural network consists of an input layer, which has the dimensionality of the input data, several hidden layers of arbitrary size, and an output layer. The idea of neural networks is to approximate a function, proposed by the input data and the expected outcome, by adjusting the weights of the interconnections between the neurons. More neurons and layers allows for the approximation of more complex functions. However, with complex architectures also the number of parameters (i.e., weights) increases that have to be optimized during training. This leads to an increased computational complexity and more data needed for training.

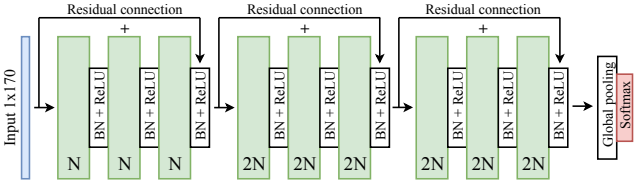
In this paper we use convolutional neural networks, i.e., a special form of FFNNs. This topology is mainly composed of three different types of layers. The main components of CNNs are (1) convolutional layers. For one-dimensional data (such as CIRs) a convolutional layer can be seen as a weighted moving average (MA) filter and the parameters that we optimize can be seen as the weights of the filter. Every convolutional operation is followed by an activation function. The idea of convolutional layers is to extract features of the preceding input data. (2) Pooling layers follow convolutional layers and apply e.g. an *avg*- or *max*-operation on the activation map to scale down the intermediate results to a lower dimension for further processing. (3) Dense layers at the end prepare the data for a classification or regression task. For our classification task (NLOS vs. LOS) we use a softmax classifier that outputs a probability for the respective classes.

In literature three different CNN models have been employed for time series classification [28]:

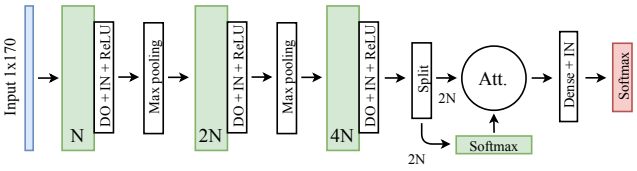
Fully convolutional network (FCN). FCNs [5] don't use local pooling, i.e., the dimensionality of the input data is kept unchanged. Additionally, the final layer before classification



(a) FCN network architecture.



(b) ResNet network architecture.



(c) Encoder network architecture.

Fig. 1: CNN architectures used for NLOS detection. (green layers: convolutional layers; BN: batch normalization; DO: dropout; ReLU: rectified linear unit).

TABLE I: Properties of the CNN architectures.

Name	# layers	# of filters	# epochs	batch size
FCN	5	$N = 2^n; n \in [1, 7]$	800	32
ResNet	11	$N = 2^n; n \in [1, 6]$	800	128
Enc	7	$N = 2^n; n \in [1, 7]$	100	12

(which is usually a fully connected layer) is replaced by a global average pooling (GAP) layer that uses the entire input data for a single pooling operation. Fig. 1a shows the architecture of this CNN. The network uses three convolutional blocks. For the first convolution a kernel size of 8 is used, followed by a kernel size of 5 and a kernel size of 3 for the last layer. For the original architecture, the number of filters is $N = 128$, whereas the architecture is also evaluated with fewer filters of $N = \{64, 32, 16, 8, 4, 2\}$. The last convolutional layer is connected to the GAP, followed by a softmax layer.

Residual network (ResNet). Residual CNNs use linear shortcuts between convolutional layers to mitigate the vanishing gradient problem [29]. This allows for deeper architectures. The ResNet we used [5], see Fig. 1b, uses 11 layers, 3 convolutional blocks, a GAP and a softmax layer as a final layer. Every convolutional block consists of 3 consecutive convolutional layers, with a kernel size of 8 in the first block, followed by a kernel size of 5 and 3 for the remaining blocks. For the original architecture, the number of filters is $N = 64$, whereas the architecture is also evaluated with fewer filters of $N = \{32, 16, 8, 4, 2\}$. Like for the FCN model, no local pooling is used, therefore the dimensions of the CIR stays constant throughout all convolutional layers.

Encoder (Enc). The encoder networks [4] (see Fig. 1c) use 7 layers, i.e., 1 input, 3 convolutional, 1 dense and 2 softmax layers. The consecutive convolutional layers have a kernel size of 5 for the first convolutional layer, followed by a kernel size of 11 and 21 for the second and third layers. For the original architecture, the number of filters is $N = 128$, whereas the architecture is also evaluated with fewer filters of $N = \{64, 32, 16, 8, 4, 2\}$. Each convolution is followed by an instance normalization operation using a parametric ReLU as activation function, a dropout operation with the rate of 0.2 and a max pooling with the length of 2. The output of the last convolution is lead into an attention mechanism [30], followed by a dense layer with an instance normalization operation and a softmax layer.

Table I summarizes the properties of the CNNs that we use in this paper. The best number of epochs and batch sizes have been determined experimentally.

IV. Experimental setup

We separate the classification in training and testing phases. First, we split the acquired data into separate training, validation, and test data. For the training of the classifiers, we use the pipeline illustrated in Fig. 2. In a preprocessing step we reduce the dimension of the CIR from approx. 1000 samples down to 170 samples by cropping it around the correlation peak and label it (i.e., NLOS or LOS) with a simple ray tracing, see Sec. IV-C. For both of the SVM classifiers/kernels, we extract features from the CIRs and we performed a grid search

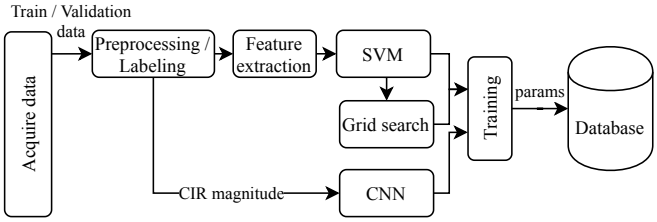


Fig. 2: Pipeline for training the classification algorithms.

using the training and validation datasets to tune the hyper-parameters (i.e., $C \in \{10^{-3}, \cdots, 10^{4}\}$, $\Gamma \in \{10^{-2}, 10^{-1}\}$). For the CNNs, the magnitudes of the CIRs are calculated and normalized as input data. To avoid overfitting to the training data, the validation datasets are used during training.

A. Hardware setup

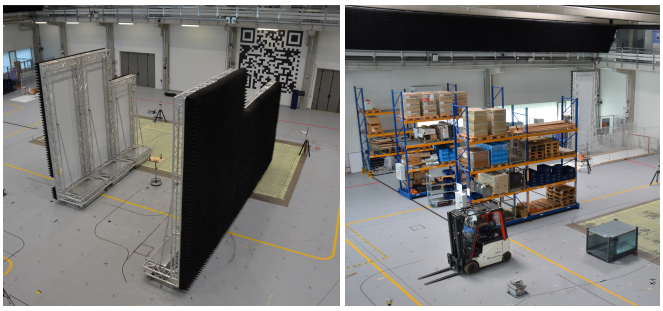
For the measurements we place six receivers at known positions and use one mobile transmitter, moving within the environment. Our tags use a decawave DW1000 module and a raspberry pi for TCP/IP communication to a central server where that data is stored. To obtain reference positions of the transmitter we use a Nikon iGPS optical reference system, which provides localization accuracies in millimeter ranges. The receivers are synchronized with a signal generator and a stationary reference transmitter is used to remove the bias from the receiver clocks.

B. Measurement campaign

Our realistic measurement campaign uses two scenarios [26] and for each of them and each receiver we record the CIRs and the ranges between the transmitters and the reference positions.

Scenario #1 uses reflecting walls that serve as a corridor (see Fig. 3a). Based on the geometric composition, propagation conditions of the radio signals can easily be labelled. Due to the receiver positions and the reflective inner surface of the walls, the data contains a high variety of propagation conditions including both NLOS and LOS links.

For an evaluation under realistic conditions, *Scenario #2* imitates an industrial environment that contains objects such as large metal shelves filled with goods, an industrial truck, a forklift, and a metal box (see Fig. 3b). The receivers are placed around the environment, which results in various propagation patterns, including shadowing, reflections and scattering.



(a) Corridor environment. (b) Industrial environment.

Fig. 3: Real-world environment of the scenarios.

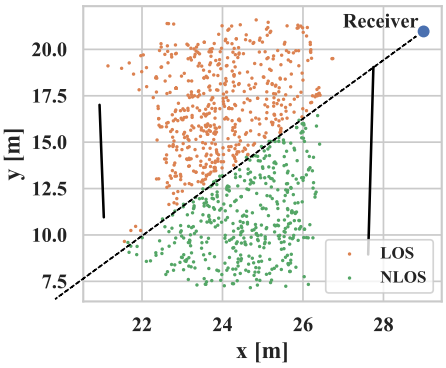


Fig. 4: Labeling the CIRs of one receiver. The scatters within the corridor show the positions of the transmitter, while the dashed line shows the transition line from LOS (orange) above the line to NLOS (green) below the line. The reflecting walls are indicated as black solid lines.

C. Labeling data

To label the data points, we use a ray tracing. Due to the simple geometrical composition of the corridor scenario, channel conditions (NLOS/LOS) can easily be simulated. For this purpose, we created a 2D model of the corridor scenario (see Fig. 4). The positions of the data is shown as scatters within the corridor and the walls of the corridor are shown as solid black lines. The dashed line shows the transition from LOS (above the line) to NLOS (below the line). However, due to the complex environment in the industrial scenario, we cannot generate labels for this scenario (and we hence evaluate the positional error in this scenario only).

V. Evaluation

Fig. 5 shows the pipeline we used for our experiments. In the test phase, the unknown test datasets are preprocessed in the same way as the training data. To measure the performance of the algorithms, we calculate the recall (i.e., true positives divided by the sum of true positives and false negatives) and the accuracy on the test data. Moreover, for an alternative evaluation, we also derive the positioning error: we use an

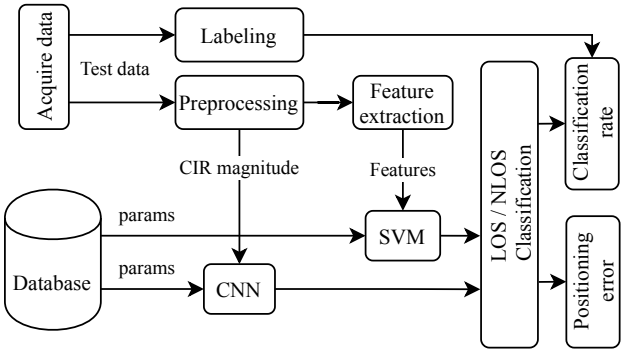


Fig. 5: Pipeline for testing the classification algorithms.

TABLE II: Data used for the experiments.

Test Scenario	Training (LOS/NLOS)	Validation (LOS/NLOS)	Test (LOS/NLOS)
Corridor all	27 / 27	6.8 / 6.8	52 / 8.5
Unknown recv.	(15-34) / (15-34)	(3.9-8.5) / (3.9-8.5)	(27-51) / (0-22)
Industrial	34 / 34	8.5 / 8.5	50

extended Kalman filter (with a constant velocity model for system dynamics) that only considers range signals classified as LOS (and that excludes NLOS data from the measurement vector) and determine the positional error (as the Euclidean distance) against the reference position of the measurements.

Table II shows three experiments that we conducted for evaluation. The first two experiments, i.e., *Cordidor all* and *Unknown recv.*, only use data from *Scenario #1*. The corridor scenario (i) is evaluated using all six receivers. We randomly selected 80 % of the data for training and validation and the remaining 20 % for testing (see Sec. V-A). We test the generalization of the classifiers to unknown receivers (ii) by training them with the data from five receivers and testing them on the data from the sixth receiver (see Sec. V-B). We test the trained models on data obtained in a different, unknown environment (iii), i.e., we train and validate the classifiers on *Scenario #1* and test them on *Scenario #2*.

A. Corridor scenario

In this experiment, we first compare the performance of the NLOS detection algorithms considering all six receivers. For this, we calculate the accuracy (the sum of true negatives and true positives divided by the total number of samples). Table III shows the results of both of the SVM baseline classifiers and our three CNN architectures (best results in bold). We can see that the CNN-based classifiers yield the highest accuracy: the Encoder model reaches (94 %), closely followed by the ResNet (93 %) and FCN (93 %). The SVM-based models both only yield lower accuracies of 84 % (linear SVM) and 83 % (radial SVM). There is also an overfitting to NLOS, as about 20 % of the LOS data is classified as NLOS and only about 10 % of the NLOS data is misclassified as LOS although we balanced the classes for training. We do not see such an imbalance with the CNN models.

To obtain further insights we also changed the number of filters for every convolutional layer and therefore the parameters used in the CNN architectures to see the influence of the network size on the classification accuracy. The results

TABLE III: Recall and accuracy for the corridor scenario.

Name	NLOS	LOS	Accuracy
LinSVM	0.89	0.83	0.84
RbfSVM	0.92	0.81	0.83
FCN	0.96	0.93	0.93
ResNet	0.96	0.93	0.93
Enc	**0.97**	**0.94**	**0.94**

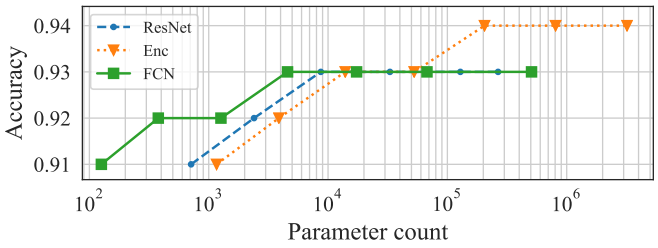


Fig. 6: Classification accuracies for the corridor scenario considering all six receivers with a reduced number of filters and thereby lower parameter counts for the CNN models.

are shown in Fig. 6. Also with the smallest version of the FCN algorithm, only using 126 parameters, we can achieve a classification accuracy of 91 %. We can see better accuracies with an increased number of parameters for all of the models. This is to be expected as the multi-stage architecture of CNNs requires a certain amount of layers and heights to configure the receptive field of the convolutional filters properly. The models can use the additional capacity in the network to account for the non-linearities in the signals.

B. Generalization to unknown environments

Next, we test for the generalization of the algorithms. For this purpose, the algorithms are trained with five receivers and tested with the unknown sixth receiver (and we train on all combinations). Different positions of the receivers lead to different reflection patterns and therefore to unknown CIRs.

Table IV shows the classification results for different receivers (e.g. column R1 shows the results of training on R2-R6 and testing on the data from R1). We see that all of the classifiers are fairly robust against different receiver positions and hardware. The SVM-based models achieve at least 72 % accuracy and the CNN models at least 81 % accuracy. In line with the results from Sec. V-A the CNN-based classifiers outperform the SVMs (with an average accuracy of 91 % and 92 % compared to accuracies of 80 % and 82 % for the SVM-based models). The influence of the network architectures is also similar to what we have seen before (and hence we omit to show the results).

C. Industrial scenario

To see the performance of the NLOS detection in a different environment, we recorded the industrial-like scenario *Scenario #2*, see Fig. 3b. Due to the different types and arrangement of objects, the CIRs may vary compared to the corridor scenario. However, the characteristic differences between NLOS and LOS signals should remain. As we could not use the ray-tracing to label our data points (due to the complex geometry) we instead use the positioning error for evaluation, i.e., we detect the NLOS/LOS of the data samples and only feed LOS samples to the EKF. Hence, as the EKF's performance is distorted by NLOS samples we see a worse performance if samples are miss-classified.

TABLE IV: Results for generalization over receivers.

	Accuracy						
Name	R1	R2	R3	R4	R5	R6	Avg
LinSVM	0.85	0.92	0.81	0.73	0.86	0.76	0.82
RbfSVM	0.83	0.89	0.82	0.72	0.84	0.73	0.80
FCN	0.94	**0.99**	**0.88**	**0.81**	0.98	0.87	**0.92**
ResNet	**0.95**	**0.99**	**0.88**	**0.81**	0.98	**0.88**	**0.92**
Enc	**0.95**	**0.99**	0.86	**0.81**	**0.99**	0.87	0.91

TABLE V: Positional errors in Scenario #2.

	Baseline	LinSVM	RbfSVM	FCN	ResNet	Enc
MAE	0.170	0.188	0.156	**0.116**	0.118	0.123
MSE	0.0530	0.0768	0.0515	**0.0215**	0.0225	0.0286

Table V shows the mean absolute error (MAE) and mean squared error (MSE) of estimated positions using the proposed NLOS detection. The EKF without NLOS detection (*baseline*) yields an MAE of 0.17 m. With the CNN-based classifiers we can even improve the positional accuracy to around 0.12 m. The low MSE indicates that there are less outliers and therefore a more robust positioning performance than with the original data. The localization accuracy, using the CNN architectures with lower size, is as good as with the original architectures.

The FCN yields the best results, which can be explained from its simple architecture, leading to a better generalization to the NLOS problem on a different environment (the low number of parameters serves like a regularization). The more complex architectures (and trainable parameters) of both the ResNet and the Encoder networks lead to overfitting to environment-specific properties of the CIRs.

Using the radial SVM, localization precision can only be improved slightly, whereas using the linear SVM achieves even a worse MAE than with the original data. However, as indicated by the cumulative distribution functions (CDFs), see Fig. 7, the error is caused by a larger amount of outliers: as the SVM approaches overfit to NLOS (as seen in Sec.V-A), larger

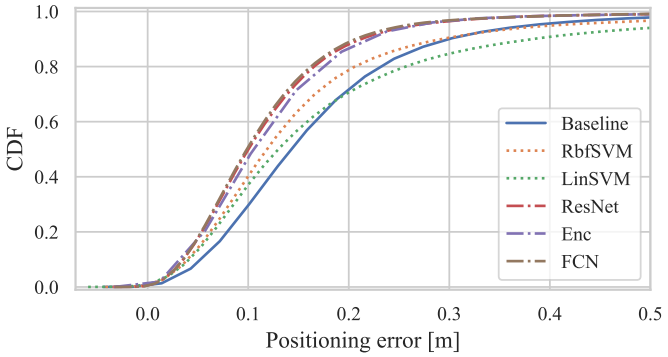


Fig. 7: CDFs of the positional errors in *Scenario #2*.

numbers of receivers are excluded, leading to a lower amount of valid information that is insufficient for positioning in some cases. In these cases, the estimated trajectories exhibited a significant drift away from the true position.

VI. Conclusion

In this paper we proposed different channel classification approaches based on UWB CIRs and CNNs. We introduced three network architectures and described in detail: FCNs, ResNets and Encoders. The proposed approaches are trained and evaluated using data from an realistic measurement campaign. A scenario with 6 receivers and a corridor of reflecting walls systematically introduces NLOS conditions, labeled using a simple ray-tracing algorithm. For this scenario, average accuracies of up to 94 % could be achieved with all of the deep learning networks, outperforming the classical, feature-based SVM approaches used as a baseline by more than 10 %. Even with data from an unknown receiver we reach 91 % average accuracy. Hence, the trained models are able to generalize to unknown datasets. Furthermore, the performance of the CNN methods only slightly changed with lower numbers of parameters, so that even networks with low complexity achieved high accuracies of over 91 %.

In a second evaluation step, the trained models were applied to a positioning scenario in an unknown environment and used to improve the positioning performance of a range-based EKF. Using the trained models to systematically exclude ranges classified as NLOS from the filter updates, the MAE could be improved by approx. 30 %. This shows that the proposed methods are also able to generalize to a new environment.

Acknowledgements

This work was supported by the Bavarian Ministry of Economic Affairs, Infrastructure, Energy and Technology as part of the Bavarian projects Leistungszentrum Elektroniksysteme (LZE) and the Center for Analytics-Data-Applications (ADA-Center) within the framework of "BAYERN DIGITAL II".

References

[1] M. Stahlke, S. Kram, T. Mumme, and J. Seitz, "Discrete positioning using uwb channel impulse responses and machine learning," in *Proc. 2019 Intl. Conf. Localization and GNSS (ICL-GNSS)*, 2019.

[2] S. Kram, C. Nickel, J. Seitz, L. Patino-Studencka, and J. Thielecke, "Spatial interpolation of wi-fi rss fingerprints using model-based universal kriging," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, Oct 2017.

[3] G. Mendoza Silva, J. Torres-Sospedra, and J. Huerta, "A meta-review of indoor positioning systems," *Sensors*, vol. 19, p. 4507, 10 2019.

[4] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," *CoRR*, 2018.

[5] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *CoRR*, vol. abs/1611.06455, 2016.

[6] M. Kolakowski and J. Modelski, "First path component power based nlos mitigation in uwb positioning system," in *2017 25th Telecommunication Forum (TELFOR)*, pp. 1–4, Nov 2017.

[7] G. İsmail, C. Chia-Chin, W. Fujio, and H. Inamura, "Nlos identification and weighted least-squares localization for uwb systems using multipath channel statistics," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 01 2007.

[8] C. Wu, H. Hou, W. Wang, Q. Huang, and X. Gao, "Tdoa based indoor positioning with nlos identification by machine learning," in *10th Intl. Conf. Wireless Communications and Signal Processing*, (Hangzhou, China), 2018.

[9] Weijie Li, Tingting Zhang, and Qinyu Zhang, "Experimental researches on an uwb nlos identification method based on machine learning," in *2013 15th IEEE International Conference on Communication Technology*, pp. 473–477, Nov 2013.

[10] Z. Zhuoqi, S. Liu, and L. Wang, "Nlos identification for uwb based on channel impulse response," in *12th International Conference on Signal Processing and Communication Systems*, (Cairns, Australia), 2018.

[11] A. Ardiansyah, G. Nugraha, H. Han, C. Deokjai, S. Seo, and J. Kim, "A decision tree-based nlos detection method for the uwb indoor location tracking accuracy improvement," *International Journal of Communication Systems*, p. e3997, 06 2019.

[12] T. Feigl, S. Kram, P. Woller, R. Siddiqui, M. Philippsen, and C. Mutschler, "A bidirectional lstm for estimating dynamic human velocities from a single imu," in *9th Intl. Conf. Indoor Positioning and Indoor Navigation*, (Pisa, Italy), 2019.

[13] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik, and C. Mutschler, "A deep learning approach to position estimation from channel impulse responses," in *Sensors*, 2019.

[14] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in tdoa-based locating systems," in *9th Intl. Conf. Indoor Positioning and Indoor Navigation*, 2018.

[15] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2016.

[16] E.-R. Jeong, J. Joung, S. Jung, and S. Chung, "Cnn-based tx–rx distance estimation for uwb system localisation," *Elect. Letters*, vol. 55, 2019.

[17] T. Feigl, T. Nowak, M. Philippsen, T. Edelhäuer, and C. Mutschler, "Recurrent neural networks on drifting time-of-flight measurements," in *9th Intl. Conf. Indoor Positioning and Indoor Navigation*, (Nantes, France), 2018.

[18] K. Cwalina, P. Rajchowski, O. Blaszkiewicz, A. Olejniczak, and J. Sadowski, "Deep learning-based los and nlos identification in wireless body area networks," *Sensors*, vol. 19, p. 4229, 09 2019.

[19] Y. Liao, Y. Hua, X. Dai, H. Yao, and X. Yang, "Chanestnet: A deep learning based channel estimation for high-speed scenarios," in *IEEE Intl. Conf. Communications*, (Shanghai, China), pp. 1–6, 05 2019.

[20] C. J. K. Y. Nguyen VH, Nguyen MT, "Nlos identification in wlans using deep lstm with cnn features," *Sensors 2018(11)*, 11 2018.

[21] Q. Liu, Z. Huang, and J. Wang, "Indoor non-line-of-sight and multipath detection using deep learning approach," *GPS Solutions*, vol. 23, 07 2019.

[22] F. Wang, Z. Xu, R. Zhi, J. Chen, and P. Zhang, "Los/nlos channel identification technology based on cnn," in *6th NAFOSTED Conf. Information and Comp. Sc.*, (Hanoi, Vietnam), pp. 200–203, 12 2019.

[23] Z. Lu, D. Wang, Y. Qin, and R. Liu, "Uwb channel classification based on deep-learning for adaptive transmission," in *12th Intl. Congress Image and Signal Processing, BioMedical Engineering and Informatics*, (Suzhou, China), pp. 1–5, 10 2019.

[24] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17429–17441, 2018.

[25] J. B. Kristensen, M. Massanet Ginard, O. K. Jensen, and M. Shen, "Non-line-of-sight identification for uwb indoor positioning systems using support vector machines," in *2019 IEEE MTT-S International Wireless Symposium (IWS)*, (Guangzhou, China), pp. 1–3, 2019.

[26] S. Kram, M. Stahlke, T. Feigl, J. Seitz, and J. Thielecke, "Uwb channel impulse responses for positioning in complex environments: A detailed feature analysis," *Sensors*, vol. 19, no. 24, 2019.

[27] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015.

[28] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[29] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th Intl. Conf. Artificial Intelligence and Statistics*, (Sardinia, Italy), pp. 249–256, 2010.

[30] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ArXiv*, vol. 1409, 09 2014.