Rain event detection in commercial microwave link attenuation data using convolutional neural networks

Julius Polz[1], Christian Chwala[1,2], Maximilian Graf[1], and Harald Kunstmann[1,2]

[1]Karlsruhe Institute of Technology (KIT), Campus Alpin, Institute of Meteorology and Climate Research (IMK-IFU), Kreuzeckbahnstr. 19, 82467 Garmisch-Partenkirchen, Germany
[2]University of Augsburg, Institute of Geography, Alter Postweg 118, 86159 Augsburg, Germany

Correspondence: Julius Polz (julius.polz@kit.edu), Christian Chwala (christian.chwala@kit.edu)

Abstract. Quantitative precipitation estimation with commercial microwave links (CMLs) is a technique developed to supplement weather radar and rain gauge observations. It is exploiting the relation between the attenuation of CML signal levels and the integrated rain rate along a CML path. The opportunistic nature of this method requires a sophisticated data processing using robust methods. In this study we focus on the processing step of rain event detection in the signal level time series of

5   the CMLs, which we treat as a binary classification problem. We analyze the performance of a convolutional neural network (CNN), which is trained to detect rainfall specific attenuation patterns in CML signal levels, using data from 3904 CMLs in Germany. The CNN consists of a feature extraction and a classification part with, in total, 20 layers of neurons and 1.4×10^5 trainable parameters. With a structure, inspired by the visual cortex of mammals, CNNs use local connections of neurons to recognize patterns independent of their location in the time-series. We test the CNNs ability to generalize to CMLs and time

10 periods outside the training data. Our CNN is trained on four months of data from 400 randomly selected CMLs and validated on two different months of data, once for all CMLs and once for the 3504 CMLs not included in the training. No CMLs are excluded from the analysis. As a reference data set we use the gauge adjusted radar product RADOLAN-RW provided by the German meteorological service (DWD). The model predictions and the reference data are compared on an hourly basis. Model performance is compared to a reference method, which uses the rolling standard deviation of the CML signal level time series

15 as a detection criteria. Our results show that within the analyzed period of April to September 2018, the CNN generalizes well to the validation CMLs and time periods. A receiver operating characteristic (ROC) analysis shows that the CNN is outperforming the reference method, detecting on average 87% of all rainy and 91% of all non-rainy periods. In conclusion, we find that CNNs are a robust and promising tool to detect rainfall induced attenuation patterns in CML signal levels from a large CML data set covering entire Germany.

20 **Keywords: precipitation, remote sensing, pattern recognition, deep learning, artificial neural network**

1 Introduction

Rainfall is the major driver of the hydrologic cycle. Accurate rainfall observations are fundamental for understanding, modeling and predicting relevant hydrological phenomena, e.g. flooding. Data from commercial microwave link (CML) networks have proven to provide valuable rainfall information. Given the high spatio-temporal variability of rainfall, they are a welcome complement to support traditional observations with rain gauges and weather radars; particularly in regions where radar is hampered by beam blockage or ground clutter. In regions with sparse rainfall observation networks, like in developing countries, CMLs might even be the only source of small scale rainfall information.

Since the work of Messer et al. (2006) and Leijnse et al. (2007) more than a decade ago, several research groups have shown the potential of CML data for hydrometeorological usage. Prominent examples are the countrywide evaluations in the Netherlands (Overeem et al., 2016b) and Germany (Graf et al., 2019), which demonstrated that CML-derived rainfall information corresponds well with gauge-adjusted radar rainfall products, except for the cold season with solid precipitation. CML-derived rainfall information was also successfully used for river runoff simulations in a pre-alpine catchment in Germany (Smiatek et al., 2017) and for pipe flow simulation in a small urban catchment in Czech Republic (Pastorek et al., 2019). A further important step was the first analysis of CML-derived rain rates in a developing country, carried out by Doumounia et al. (2014), with data from Burkina Faso.

In general, the number of CMLs available for research has increased significantly over the last years and researchers from several countries have gained access to CML attenuation data. Currently, data from 4000 CMLs over Germany is recorded continuously with a temporal resolution of one minute via a real-time data acquisition system (Chwala et al., 2016). The number of existing CMLs over Germany is 30 times higher (Bundesnetzagentur, 2017), amounting to 130.000 registered CMLs. Consequently, it is envisaged to increase the number of CMLs included in the data acquisition.

With this large number of CMLs available in Germany and with new data being retrieved continuously, there is a need for optimized and robust processing of these big data sets. Several studies address the details of the processing steps which are required for deriving rainfall information from CMLs. These steps involve, e.g. the detection of rain events in noisy raw data, the filtering of artifacts, correcting for bias due to wet antenna attenuation (WAA) and the spatial reconstruction of rainfall fields. Uijlenhoet et al. (2018) give a general overview of the required processing steps and the existing methods and Chwala and Kunstmann (2019) discuss and summarize the related current challenges.

The first of these processing steps, called rain event detection, is the separation of rainy (wet) and non-rainy (dry) periods. It is challenging, because CML signal levels can show high fluctuations, even when there is no rain, e.g. due to multi-path propagation (e.g. Chwala and Kunstmann, 2019, Fig. 6). Therefore, the main difficulty is to distinguish between noise and signal fluctuations caused by light rain along the CML path. After successfully detecting rain events, an attenuation baseline is determined and actual rain rates can be derived via the k-R power law which relates specific attenuation k in dB km^{-1} to rain rate R in mm h^{-1}. Misclassifications of wet and dry periods lead to over- or underestimation of rainfall. Therefore, it is important to optimize the rain event detection as an isolated processing step first and to optimize subsequent processing steps afterwards.

So far, several methods for rain event detection with CMLs have been proposed. Schleiss and Berne (2010) introduced a threshold for the rolling standard deviation (RSTD) of the attenuation time-series as a criteria to detect rain events. Overeem et al. (2011) introduced the 'nearby link approach', where a period is considered wet if the increase of CML specific attenuation correlates with the attenuation pattern of nearby CMLs. They concluded that this is only applicable for dense CML networks

60 with a high data availability. Chwala et al. (2012) introduced Fourier transformations on a rolling window of CML signal levels to detect the pattern of rain events in the frequency domain. Wang et al. (2012) used a Markov switching model. Kaufmann and Rieckermann (2011) have shown the applicability of random forest classifiers and Gaussian factor graphs.

At the same time, deep learning is a rapidly evolving field that is becoming increasingly popular in the earth system sciences. A large field of application is remote sensing using artificial neural networks for image recognition (Zhu et al., 2017). Deep

65 learning is also an established method in time-series classification (Fawaz et al., 2019). In both articles, convolutional neural networks (CNNs) are considered one of the leading neural network architectures for image and time-series classification. CNNs are inspired by the visual cortex of mammals and they are designed to recognize objects or patterns, regardless of their location in images or time-series (Fukushima, 1980). They are characterized by local connections of neurons, shared weights and a large number of layers of neurons, involving pooling layers (LeCun et al., 2015). CNNs with one dimensional input data

70 (1D-CNNs) have already been used for time-series classification, e.g. for classifying environmental sounds (Piczak, 2015). This makes 1D-CNNs a promising candidate for the task of rain event detection in CML signal levels.

Other artificial neural network architectures have already been proposed for rain event detection. Đorđević et al. (2013) used a simple Multilayer Perceptron (MLP) with data from a single CML. Habi and Messer (2018) tested the performance of Long Short-Term Memory (LSTM) networks to classify rainy periods from 15 minute Min-Max values of CML signal levels for

75 34 CMLs. Kim and Kwon (2018) used LSTM networks on instantaneously sampled signal levels from 10 CMLs, which are situated close to each other, at a temporal resolution of 15 seconds.

All rain event detection methods have to make a similar trade-off: A liberal detection of wet periods is more likely to recognize even small rain rates, while it will produce more false alarms during dry periods. On the other hand, a conservative detection will accurately classify dry periods, but is more likely to miss small rain events. One can address this by two means. First, by

80 increasing detection rates on both wet and dry periods as much as possible and therefore decreasing the impact of the trade-off. Second, by allowing the flexibility to easily adjust the model towards liberal or conservative detection, e.g. by only changing a single parameter.

Until now, there have been few studies analyzing the performance of rain event detection methods on large data sets. Overeem et al. (2016a) tested the nearby link approach using 2044 CMLs distributed over the Netherlands with a temporal coverage of

85 2.5 years of data. In Graf et al. (2019) we adjusted the RSTD method to one year of data from 3904 CMLs to set a benchmark performance on this data set. By optimizing thresholds for individual CMLs we explore the full potential of the RSTD method for this data set, yielding good results for the warm season with liquid precipitatoin. While the RSTD method is simple to implement and has only two parameters to optimize, it is limited to measuring the amount of fluctuations, rather than the specific pattern. More room for optimization is expected using machine learning techniques for pattern recognition. Since the variety

90 of signal fluctuations and possible sources of error rises with large quantities of CMLs, it has to be proven that artificial neural

networks allow for high-performance, fast and robust processing of large data sets, i.e. high variability of frequency, length and spatial distribution of the analyzed CMLs and high variability of rain rates and event duration for a large amount of analyzed periods.

The objective of this study is to evaluate the performance of 1D-CNNs to detect rainfall induced attenuation patterns in in-
95   stantaneously measured CML signal levels. We test the CNNs ability to generalize to new CMLs and future time periods. To validate our results, we compare the CNN to the method of Schleiss and Berne (2010) using a large data set consisting of six months of data from 3904 CMLs distributed over entire Germany.

2 Methods

The following definition of rain event detection with CMLs is the basis of our methodology: Rain event detection is a binary
100   classification problem. Given a time window $X_{t,w,i}$ of CML signal data, where t is the starting time, w is the window length and i is the index specifying a unique CML path, we have to decide if there is attenuation caused by rain (wet) or not (dry). A time window is assigned the label 1 if it is wet or 0 if it is dry. The available information to do this classification depends on the used data acquisition and on which information is provided by the CML network operator. In the following, we describe how a CNN can be used as a binary classifier to succeed in this task.

105   **2.1 Data set**

We use a CML data set that has been collected in cooperation with Ericsson Germany through our custom CML data acquisition system Chwala et al. (2016). It covers 3904 CMLs across entire Germany. The CML path length ranges from 0.1 km to more than 30 km, with an average of around 7 km. CML frequencies range from 10 to 40 GHz. The acquired data consists of two sub-links per CML, transmitting their signal in opposite directions along the CML path. For each sub-link a received signal
110   level (RSL) and a transmitted signal level (TSL) is recorded at a temporal resolution of 1 minute and a power resolution of 0.3 dB for RSL and 1.0 dB for TSL. The recorded period used in this study starts in April 2018 and ends in September 2018, to focus on the periods which are dominated by liquid precipitation, where CMLs perform better than during the cold season (Graf et al., 2019). The data is available at 97.1% of all time steps and gaps are mainly due to outages of the data acquisition system.

115   As reference data we use the gauge adjusted radar product RADOLAN-RW provided by the German meteorological service (DWD). It has a spatial resolution of 1x1 km, covering entire Germany on 900x900 grid cells. The temporal resolution is 60 minutes and the resolution for the rain amount is 0.1 mm (Winterrath et al., 2012). To compare to this reference, the window length w is set to 60 minutes and therefore w is omitted in the notation below. Along each CML i, the path-averaged mean hourly rain rate $R_{t,i}$ is generated from the reference, using the weighted sum

120   $$R_{t,i} = \frac{\sum_k l_{k,i} r_{k,t}}{l_i}, \qquad (1)$$

where k is indexing the RADOLAN grid cells intersected by the path of i. The rain rate of each grid cell is $r_{k,t}$. Furthermore, $l_{k,i}$ is the length of the intersect of k and i and l_i is the total length of i. A time window $X_{t,i}$ is considered wet if $R_{t,i} \geq 0.1$ mm h^{-1} and dry otherwise.

2.2 Pre-processing

125 Before training and testing an artificial neural network, the raw time-series data has to be pre-processed. We do this to sample time windows of a fixed size, which are normalized and labelled according to the reference.

First, the full data set, consisting of all available CMLs, is split into three subsets. One subset is used for training the CNN (TRG), one is used for validation and to optimize model hyper-parameters (VALAPR) and one is used for testing only (VALSEP). The data set TRG consists of data from 400 randomly chosen CMLs in the period from May to August 2018.

130 VALAPR covers the remaining 3504 CMLs during April 2018 and VALSEP consists of data from all 3904 CMLs during September 2018.We used this splitting routine to avoid information leakage from the training to the validation data. There can be a high correlation of signal levels between CMLs that are situated close to each other (Overeem et al., 2011). Therefore, the measurements contained in VALAPR or VALSEP can not be taken from the same time range as for TRG. Using only 10% of all available CMLs for training allows us to analyze the CNNs generalization to the remaining CMLs in the validation data set.

135 No CMLs were excluded from the analysis.

For each of the two sub-links of a CML, we compute a transmitted minus received signal level (TRSL). Within one TRSL time-series, randomly occurring gaps of up to three minutes of missing data are linearly interpolated. Here, we assume that the temporal variability of rainfall is not high enough such that entire rain events can be hidden in such short gaps. The next step is to normalize the data. Normalization of training and validation data is a commonly used procedure in deep learning to enhance

140 the model performance. We perform the normalization as a pre-processing step and outside the CNN. After testing various normalization techniques it turned out that the best performance of the CNN can be achieved by subtracting the median of the preceding 24 hours from each time step.

The set of starting time-stamps of the hourly reference data set is denoted T_{rad}. For each CML i and each starting time $t \in T_{rad}$ a sample of data $\bar{X}_{t,i}$ is composed from 180 minutes of TRSL from the two sub-links starting at $t-120$. The first 120 minutes

145 serve as a reference to previous behaviour of the same CML and the last 60 minutes are the period $X_{t,i}$ that has to be classified. An example TRSL over a period of 8 days is given in Fig. 1 (a).

After gap filling we exclude all samples with missing values from the analysis. Since we loose three hours of data whenever there is a gap, the interpolation routine increases the number of available samples from 79.9% to 95.4%.

To train the CNN we have to balance the wet and dry classes in the data set (Hoens and Chawla, 2013). The under-sampling

150 approach to achieve an equalized (50:50) class ratio is to randomly discard samples of the majority class, i.e. dry samples. This approach is chosen since we assume that dry periods mostly consist of redundant samples with only small fluctuations. Later, we check that there is no loss in performance by evaluating the unbalanced data. The initial percentage of wet samples is between 5-7%. We perform the balancing on TRG and VALAPR. The balanced version of VALAPR is denoted VALAPRB. VALAPR and VALSEP are kept as unbalanced data sets for validation. TRG already denotes the balanced data, since the orig-

Figure 1. Performance of the CNN and the reference methods for an example CML time-series. Predictions from the CNN yield an MCC (e) of 0.57. Predictions through σ_{opt} (c) and σ_{q80} (d) yield MCCs of 0.47 and 0.33 Note that the TRSL and RSTD time series of sub-link 2 are almost identical to those of sub-link 1 and are shown in light grey.

155 inal unbalanced training data set is not used in the analysis. In total, the number of samples is 7×10^4 for TRG, 2.9×10^5 for VALAPRB, 2.35×10^6 for VALAPR and 2.72×10^6 for VALSEP.

2.3 Neural Network

CNNs especially apply to time-series classification when patterns have to be recognized in longer sequences of data but the location of the occurring patterns is variable. They are therefore suitable classifiers for sensor data like the TRSL from CMLs.

160 The expected advantage of the CNN over the reference method is that it is able to recognize the rainfall specific patterns, rather than just the amount of fluctuations. Like other neural network architectures they consist of a series of layers of neurons (Fig. 2). The first layer receives the input data and the last layer serves as an output for a prediction. The hidden layers in between are organized in two functional parts. The first part consists of a series of convolution and pooling layers and is used

to extract features from the raw model input. Earlier convolution layers identify simple patterns in the data, which are used to identify more complex patterns in subsequent layers. The second part consists of fully connected layers of neurons and is used to classify the input based on the features extracted by the convolutional part.

Before a CNN can be used as a classifier, it has to be trained on data in a supervised learning process. All layers have a set of trainable parameters, so called weights, which are optimized during the training process according to a learning rule. To be able to monitor the model performance, a test data set is evaluated regularly during the training process. Training is stopped before the model starts to over-fit, i.e. the performance on the test data set either stagnates or drops, while it still rises for the training data.

2.3.1 Network architecture

We use a 1D-CNN, which has the same structure as the basic 2D-CNN, with alternating convolutional and pooling layers followed by fully connected layers. The only difference is that the input data of the convolutional layers is one dimensional. The specific architecture and parameterization was optimized experimentally. To give an intuitive description of our CNN, we follow the approach provided in (LeCun et al., 2015, p. 439):

The convolutional part of the CNN consists of four blocks of two convolutional layers followed by a max pooling layer and one block of one convolutional and one average pooling layer (see Fig. 2). Convolutional layers extract feature maps by passing local patches (3x1) of input from the preceding layer through a set of filters followed by a rectified linear unit. Each filter creates a different feature map. The pooling layer then combines semantically similar features by taking the maximum (resp. average) within one local patch. This way, the dimension of the input is gradually reduced while, at the same time, the number of extracted features increases.

The fully connected part of the CNN consists of two layers with 64 neurons each and an output layer with one neuron. Its output is a prediction between zero and one, that can be interpreted as the likeliness for the input sample to be wet or dry. To avoid over-fitting to the training data dropout layers are added with a dropout ratio of 0.4 (Srivastava et al., 2014).

We implement the CNN in a Python framework using the Keras (version 2.2.4) backend for Tensorflow (version 1.12.0) (Chollet, 2015; Martín Abadi et al., 2015). For the model architecture, type, number and order of layers has to be chosen. There are several hyper-parameters that can be specified in the model setup. Each layer has a number of hyper-parameters that can be adjusted, e.g. the size of the local patch or the number of filters in a convolutional layer. We optimized all hyper-parameters iteratively by evaluating the performance of several reasonable configurations on the test data set VALAPRB, and by choosing the model with the best performance metrics (see 2.4). The final CNN consists of 20 functional layers with a total of 140,033 trainable parameters. The organization of those layers is shown in the network graph in Fig. 2. The detailed model and training specifications, all hyper parameters and the weights of the trained CNN can be retrieved from the code example at https://github.com/jpolz/cnn_cml_wet-dry_example.
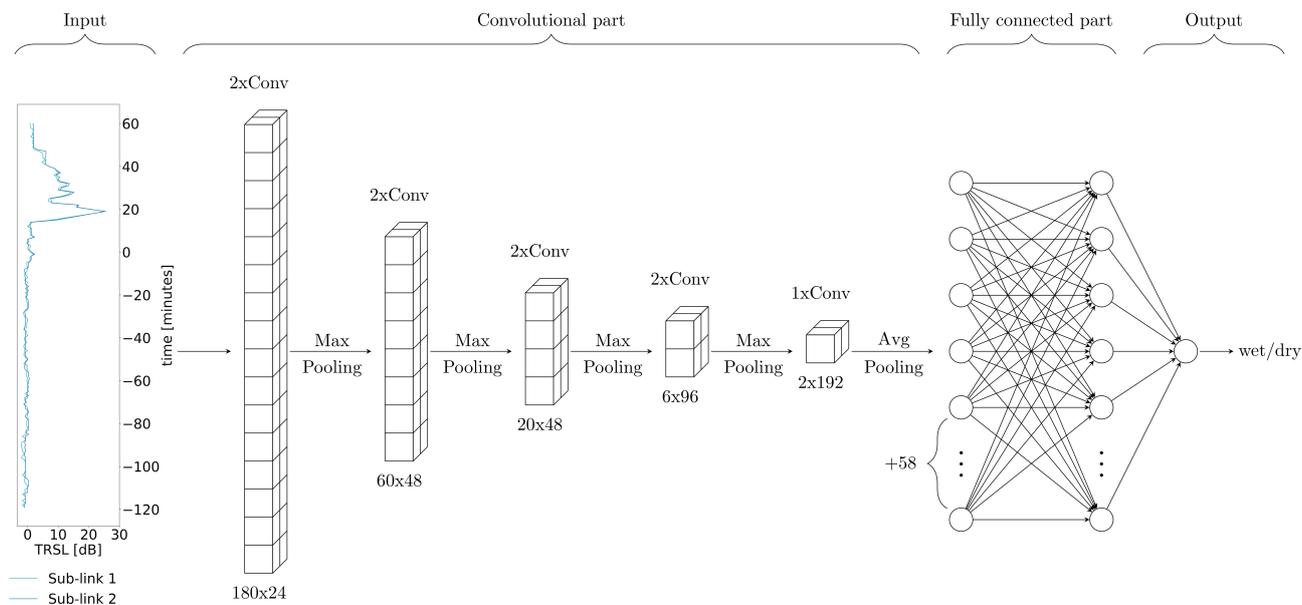
Figure 2. Graphical illustration of the CNNs architecture. The Input shows one sample $\bar{X}_{t,i}$ of data consisting of 180 minutes of TRSL from the two sub-links of one CML. Convolutional and pooling layers reduce the input dimension from 180 to 2, while a total of 192 features are extracted. Numbers below convolutional layers are the layer output dimensions, i.e. input dimension times the number of filters. The size of the local patch in a convolutional layer is 3. Based on the extracted features, the fully connected layers predict a class, which is stored in the output layer.

2.3.2   Training setup

CNNs are feed-forward neural networks, which are trained by a supervised learning algorithm (Goodfellow et al., 2016). Batches of samples are passed through the network and the outputs are compared to the reference labels. After each batch a loss function is computed and the weights are updated according to a learning rule. Here, the learning rule is stochastic gradient

200   descent with binary cross-entropy as a loss function and an initial learning rate of 0.008 (Bottou et al., 2018). The training data set TRG consists of 7 batches with 10^4 samples each and the validation data set is VALAPRB. One training epoch is finished when the whole data set is used once. After each epoch the training and validation data sets are evaluated and the learning rate is decreased slightly.

The training is stopped once the loss function of the validation data set starts to increase or does not significantly decrease

205   within multiple epochs. The model is then considered ready for classification. The final number of epochs was 2000, since after 1500 epochs the validation loss did not decrease and the accuracy was increased only by 0.0005, while, at the same time, the loss of the training data decreased by 0.02 (see Fig. 5 (a)). On one Nvidia Titan Xp GPU the training time was 1.5 hours.

Classifying 3904 samples, i.e. one time-step for all CMLs, took 200ms. For further verification, we repeat the training multiple times with a different randomization (selection of CMLs and balancing) of TRG and VALAPRB.

210 2.4 Validation

Our CNN is a probabilistic classifier. The raw model output $\bar{Y}_{t,i}$ is on a continuous scale from 0 to 1 (see Fig. 3), representing the estimated likeliness that a sample $\bar{X}_{t,i}$ is wet. A threshold $\tau \in [0,1]$ is then set to decide whether a sample is wet or not, leading to the prediction rule

$$\tilde{Y}_{t,i} = \begin{cases} 1, & \text{if } \bar{Y}_{t,i} > \tau \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

215 Classification results are compared to the reference in a confusion matrix, shown in Table 1, which is the basis for computing further metrics. The normalized version of the confusion matrix consists of the occurrence rates of TP, FP, FN and TN samples, defined as

$$TPR = \frac{TP}{TP + FN}, \qquad (3)$$

220 $$FPR = \frac{FP}{FP + TN}, \qquad (4)$$

$$FNR = \frac{FN}{TP + FN}, \qquad (5)$$

and

$$TNR = \frac{TN}{FP + TN}. \qquad (6)$$

225 As a first metric for validation we use the accuracy score, defined as

$$ACC = \frac{TP + TN}{\text{total population}} \in [0,1]. \qquad (7)$$

It is a measure for the percentage of correct classifications being made. It is dependent on the class balance of the data set. The balance of wet and dry samples in the data set is directly related to the regional and seasonal climatology. Therefore, this metric is not climatologically independent.

230 The second metric we use is the Matthews correlation coefficient (MCC), also known as ϕ-coefficient, which is a commonly used metric for binary classification (Baldi et al., 2000). It is acknowledging the possibly skewed ratio of the wet and dry periods and is high only if the classifier is performing good on both of those classes. It is defined as

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \in [-1,1], \qquad (8)$$
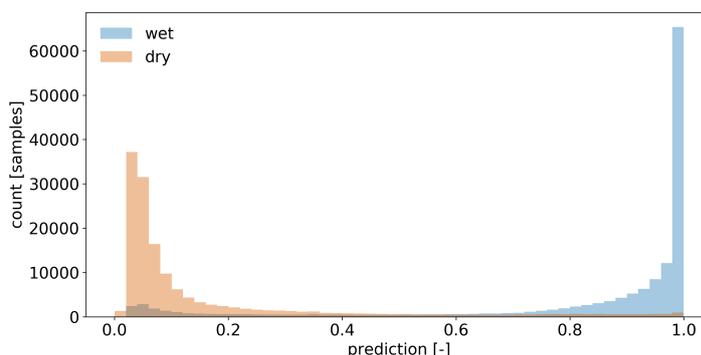
Figure 3. Raw CNN predictions on VALAPRB, coloured according to the reference.

Table 1. Confusion matrix

		reference	
		wet	*dry*
prediction *wet*		true wet (TP)	false wet (FP)
prediction *dry*		missed wet (FN)	true dry (TN)

where an MCC of 0 represents random guessing and an MCC of 1 represents a perfect classification. A strong correlation is
235 given at values above 0.25 (Akoglu, 2018).

The third metric we use is the receiver operating characteristic (ROC), defined by the pair $(FPR, TPR) \in [0,1] \times [0,1]$
(Fawcett, 2006). The domain of the ROC is called ROC space. The point (0,1) represents a perfect classifier, while the
[(0,0),(1,1)] diagonal represents random guessing. The ROC is independent of the ratio of wet and dry periods and there-
fore a climatologically independent measure for the classifier's performance on rain event detection. Each $\tau \in [0,1]$ leads to
240 a ROC resulting in a ROC curve $\gamma \subset [0,1] \times [0,1]$ (e.g. Fig. 4). The performance of a classifier for different values of τ is
measured by the area

$$AUC = \int_0^1 \gamma d\tau \in [0,1] \qquad (9)$$

under the ROC curve. Since changing τ directly influences the prediction rule (Eq. 2), it can be adjusted causing the model
to classify in a conservative (below [(0,1),(1,0)] diagonal in ROC space) or liberal (above diagonal) manner. We can therefore
245 address the trade-off between true wet and true dry predictions as mentioned in the introduction. This way, the AUC becomes a
measure of the flexibility of a classifier, i.e. the ability to show good performance with a more conservative or liberal threshold
τ.
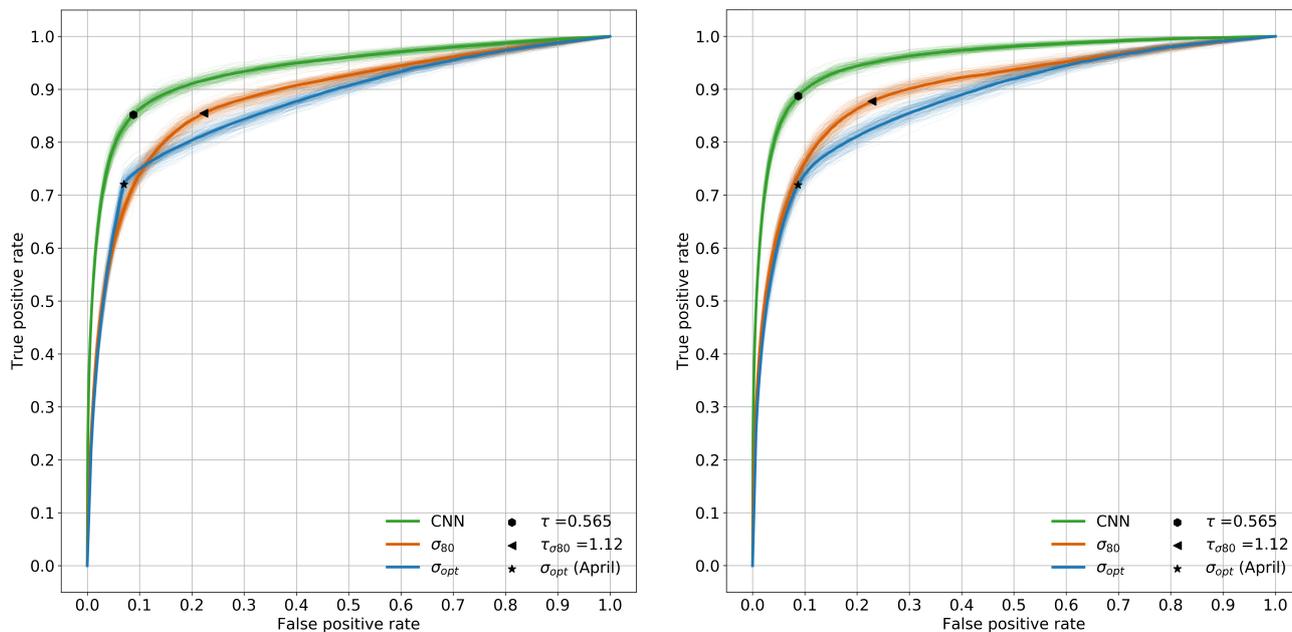
Figure 4. Receiver Operating Characteristic curves on VALAPR (left) and VALSEP (right). Fine lines are generated by 200 random selections (bootstrapping) of 1% of the samples and account for the variability of the model performance during a random short period ($\sim$ eight hours) of data.

2.5 Reference method

To be able to compare the performance of the CNN to previously used methods for rain event detection we implement a
250   reference method. We choose the method introduced by Schleiss and Berne (2010) which we previously used for processing and validation of CML-derived rain rates for one year of CML data in Germany (Graf et al., 2019). It is based on the following assumption: The standard deviation values of fixed-size windows of TRSL is bounded during dry periods, whereas it exceeds this boundary during wet periods and therefore allows for distinguishing the two classes. This assumption has proven to give good results on our data set, however there are known drawbacks. The method is limited to measuring the amount of signal
255   fluctuations and there are multiple effects that can cause high signal fluctuations during dry periods. Some of the factors are known, like multi-path propagation, but others are unknown and still need to be investigated.

The method is applied by computing a rolling standard deviation of the TRSL time-series. The normalization step is not necessary for this method. The window length is 60 minutes and the standard deviation value is written to the timestamp in the center of this window. A period $X_{t,i}$ is considered wet if at least one standard deviation value on one or both sub-links exceeds
260   a threshold σ.

We compare two different thresholds σ, which are computed individually for each CML. The first one, denoted σ_{80}, is the 80th percentile of the 60-minute rolling standard deviation of one month for a certain CML multiplied by a scaling factor

Table 2. Performance metrics of rain event detection methods on VALAPR and VALSEP

	Method	TPR	TNR	ACC	MCC	AUC
VALAPR	CNN	**0.85**	0.91	0.91	**0.54**	**0.94**
	σ_{q80}	0.85	0.78	0.78	0.35	0.89
	σ_{opt}	0.72	**0.93**	**0.92**	0.50	0.87
VALSEP	CNN	**0.89**	**0.91**	**0.91**	**0.52**	**0.96**
	σ_{q80}	0.88	0.77	0.78	0.32	0.90
	σ_{opt}	0.72	**0.91**	0.90	0.42	0.88

which is constant for all CMLs. In our case, the threshold is computed for VALAPR in April and VALSEP in September. The scaling factor of 1.12 is adopted from Graf et al. (2019). The second one, denoted σ_{opt}, is optimized against the reference

265 by maximizing the MCC. We computed it for April 2018 and then reapplied it to September 2018 to test its transferability to future time periods. To derive ROC curves, we applied a scaling factor τ_σ to each of the standard deviation thresholds. In the following we will refer to σ_{80} and σ_{opt} as both the resulting detection method and the threshold.

3 Results

During training on TRG, the performance of the CNN was evaluated on VALAPRB after each epoch. The resulting graphs

270 of loss, ACC, TPR and TNR during the training process are shown in Fig. 5. For all three variables the performance on TRG and VALAPRB were similar across all epochs with slightly higher performance on TRG. The threshold τ was optimized using VALAPRB, by maximizing the MCC, with a resulting value of τ=0.565. As shown in Fig. 3 the sensitivity for small changes of τ is not very high around its value of 0.565. A final evaluation on VALAPRB led to a TPR of 0.85 and a TNR of 0.91. No significant changes in the training process or in the resulting performance could be observed with different randomizations of

275 TRG and VALAPRB.

We evaluated the performance of the CNN and both reference methods using the unbalanced data sets VALAPR and VALSEP. The complete list of the achieved performance metrics is presented in Table 2. Applying the threshold τ to the CNN predictions yielded TPRs of 0.85 (VALAPR) and 0.89 (VALSEP) and TNRs of 0.91 (VALAPR) and 0.91 (VALSEP) (see also Fig. A1). On average, 9% of the dry periods were falsely classified as wet and 13% of the wet periods were missed. With a scaling factor

280 $\tau_{\sigma_{q80}}$ of 1.12, a similar TPR as with the CNN was achieved. But on both VALAPR and VALSEP the TNR of the σ_{q80} method was substantially lower than the CNNs TNR. On the other hand σ_{opt} achieved similar TNRs than the CNN but at the cost of lower TPRs.

For both data sets, the CNN's ROC showed a higher TPR for any fixed FPR than the reference methods (see Fig. 4). As a consequence, the AUC was largest for the CNN. On VALAPR, σ_{opt} yielded a better ROC than σ_{q80}, but only for low FPR

285 values. On VALSEP σ_{q80} achieved a better ROC than σ_{opt}. The ROC curves of the CNN and σ_{q80} had a very similar convex
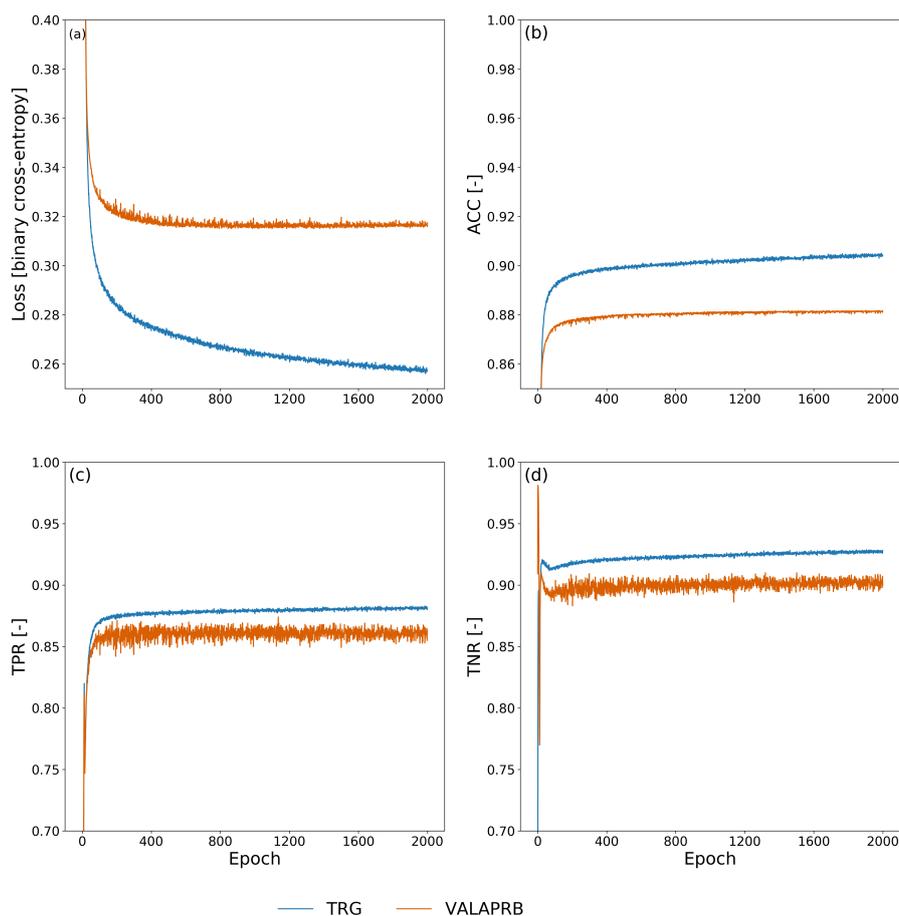
Figure 5. Statistics of variables that were monitored during the training process.

shape. Compared to the other two curves the ROC curve of σ_{opt} showed a higher asymmetry. The CNN achieved the highest ACC and MCC scores with an average of 0.91 and 0.53 on both data sets. While σ_{opt} has the second highest ACC and MCC scores, the area below the ROC curve is lowest for both data sets.

We compare the ACC on detecting samples with a specific RADOLAN-RW rain rate of $x < R_{t,i} < x + 0.1$ in Fig. 6. From all rain events where $R_{t,i} > 1.6\ mm$ 99.4% were correctly detected by the CNN. On the other hand around 36.4% of all rain events with $R_{t,i} < 0.2\ mm$ were missed. All three methods have a lower ACC, the lower the rain rate is. While σ_{q80} shows an ACC for wet periods of different rain intensities, that is very similar to that of the CNN, σ_{opt} misses more small events. On the other hand σ_{q80} is producing more false wet classifications than the CNN and σ_{opt}.
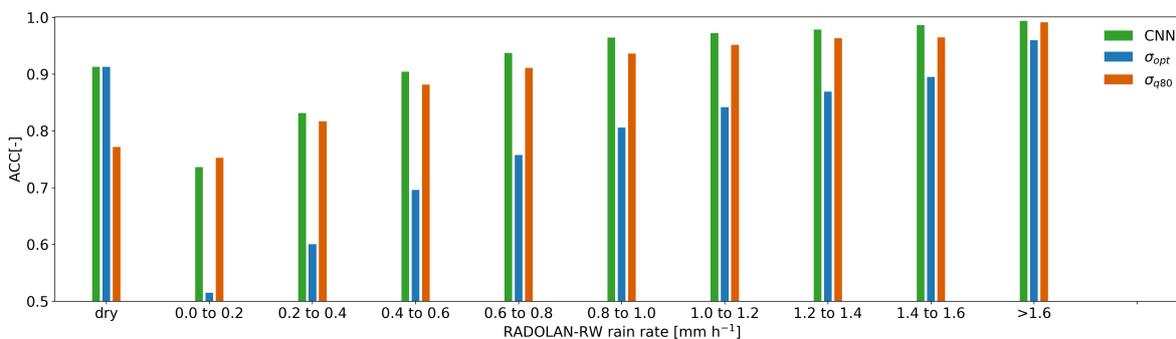
Figure 6. Each bar shows the ACC score on samples from VALSEP, grouped by the reference rain rate. The lower ACC limit of 0.5 on the y-axis represents random guessing.
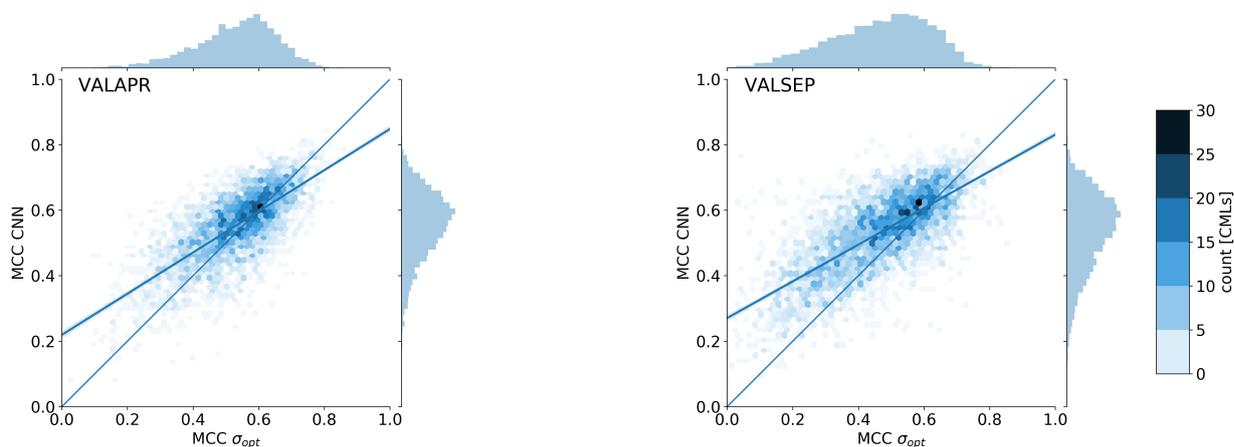
Figure 7. Scatter density plots of the MCC achieved by the CNN and σ_{opt} on data from individual CMLs. While the MCC optimized method σ_{opt} achieves comparable MCC values in April, where it was optimized, the performance drops in September.

The MCC was computed individually for each CML and each validation data set. Figure 7 shows scatter density plots comparing the individual MCC scores of the CNN and σ_{opt}. The CNN's MCC on VALAPR is higher for 61.7% of all CMLs and on VALSEP it is higher for 73.4% of all CMLs.

4 Discussion

4.1 Performance

We evaluate the performance of the CNN to detect rain events by two means. First, we compare it to the performance of
300 a reference method. Second, we estimate if the model is performing in a near optimal state or if we expect that a higher
performance could be achieved. The comparison to the results of previous studies, e.g. Overeem et al. (2016a), is difficult since
the performance is depending on the intensity of rain events (see Fig. 6) and since there is a large variability of performance
within the CMLs (see Fig. 7).

Since the results on both validation data sets are very similar (see Table 2) we further focus on VALSEP, which was not used to
305 optimize the model hyper-parameters. With an ACC of 0.91 and an MCC of 0.54 the correlation between the CNN predictions
and the reference data set RADOLAN-RW can be considered as very high.

The CNN and the reference method σ_{opt} have the same ACC value. At the same time the CNN's MCC is 0.1 points higher,
despite the fact that σ_{opt} is MCC optimized for each CML. The high ACC of σ_{opt} is due to the high TNR and the fact that 95%
of all samples are negative (dry). At a similar ACC and TNR we could increase the TPR, or rain event detection rate, by 0.17.
310 This constitutes a major improvement by the CNN. As shown in Fig. 7 the improvement is higher for CMLs with lower MCC,
making the whole CML data set more balanced in performance and therefore more trustworthy for quantitative precipitation
estimation.

The CNN's improvement in ACC and MCC over σ_{q80} was even higher with 0.13 and 0.2. While the TPR of σ_{q80} and the
CNN are similar, the TNR is much lower for σ_{q80}. Thus the CNN shows substantial improvement in correctly classifying dry
315 periods.

While the RSTD method can be set up to either have a high TPR (σ_{q80}) or a high TNR (σ_{opt}), the CNN achieves both rates at
the same time. Thus, the CNN shows a better overall performance than the reference methods and therefore improves on the
trade-off as mentioned above. This general observation is illustrated by the example time-series in Fig. 1, which shows a CML
with an average MCC (achieved by the CNN) of 0.57.
320 All three methods have limitations to detect events with rain rates smaller than 0.3mm. This is likely due to the detection limit
of CMLs in our data set which is in the same range. The detection limit depends on frequency, length and signal quantization
of a CML. For example, at a frequency of <20 GHz and at a length of <10 km a path averaged rain rate of 1 mm h^{-1} creates
a maximum of 1 dB of attenuation (Chwala and Kunstmann, 2019, Fig. 7). In some cases the quantization (0.3dB for RSL and
1dB for TSL) might therefore not allow for a detectable signal.
325 Differences in the performance on VALAPR and VALSEP can be traced back to a different distribution of occurring rain rates.
While in April 35.5% of all events are in the critical range from 0.1mm to 0.3mm, there are only 32% in September. In both data
sets the performance on higher rain rates (> 1.6 mm) and dry periods is almost identical. Therefore the loss of performance in
April is due to the slightly worse performance of the CNN on smaller rain rates which occur more often in VALAPR than in
VALSEP.
330 It should not be expected that the rain events detected through CMLs and the events detected by the radar coincide completely.

Both methods produce artifacts that are mistaken as rainfall, or they miss events due to their detection limits. From all false classifications that the CNN makes on VALSEP there are 50% with a raw model output between 0.2 and 0.8. Here the CNN does not give a certain prediction. This is due to very similar signal patterns in noisy dry periods and small rain rates. The other 50% of those samples are, according to the CNN, very likely to belong to the falsely predicted class. Despite this being an issue

335 for many CMLs about 10% have a ROC of $(> 0.97, < 0.1)$ and correlate very well with the RADOLAN reference. Therefore, we expect that less errors could be made when training with a perfect reference data set, but there would still be errors due to artifacts or insensitivity in CML measurements.

Despite those errors, which occur mostly for small rain rates, the correlation of wet and dry periods between RADOLAN-RW and our CML data set is very high. The performance boost in rain event detection gained through the CNN is very promising

340 for future applications in quantitative precipitation estimation with CMLs.

4.2 Robustness

The CNNs ability to generalize to previously unknown CMLs is very high. As seen in the training results the learning curves for both training and validation show a similar dynamic (see Fig. 5). As expected the training data showed better performance, but the validation was close at all epochs.

345 Only 10% of all available CMLs were used for training. The remaining 90% were only used to prevent the model from overfitting to the training data and to choose the model architecture and to slightly adjust τ. Thus no information about the validation data was given directly to the model. The resulting model architecture and hyper-parameters are not specific enough to store this information. The high performance in ACC, MCC and ROC on data set VALAPR, together with the learning curves in Fig. 5), therefore prove that the CNN was able to recognize the attenuation pattern in the signal levels of a large number of

350 previously unknown CMLs.

The stability of the CNNs performance for future time periods is analyzed using the results on VALSEP. While the training was done with TRG including the period of May to August 2018, the performance in September was similar. Compared to the results on VALAPR the CNN shows even higher performance on VALSEP, which can be explained by the lower percentage of samples with small rain rates in September, which are challenging to classify (see Fig. 6 and A2). When we compare the CNNs

355 accuracy per rain rate between VALAPR and VALSEP, we see that there are no major differences in the individual scores. Therefore the method can be considered as very stable throughout the analyzed time period, while differences in overall performance mostly stem from different distributions of the occurring rain rates. The reference method σ_{opt}, which was optimized in April, loses performance in September, where it is outperformed by the adaptive method σ_{q80}. The bootstrapping in Fig. 4 shows that all three methods perform almost equally well on small random subsets of the validation data. The CNN shows the

360 lowest variability.

As a measure for the flexibility of a classifier we adopted the ROC analysis in section 2.4. A model is called flexible if it has a high area below its ROC curve and if the curve is axis-symmetric with respect to the [(0,1),(1,0)] diagonal of the ROC space. As observed both the CNN and σ_{q80} show a symmetrical ROC curve. Therefore they perform almost equally well with a liberal or conservative threshold with a slight tendency to the conservative side. On the other hand σ_{opt} shows a skewed performance,

365 with a strong tendency to the conservative side. The area AUC below the ROC curve was highest for the CNN, making it the most flexible classifier. We can adjust τ for a ROC of either $(0.03, 0.7)$ or $(0.3, 0.94)$ and a smooth, concave transition in between (see Fig. 4).

We conclude that within the analyzed period the CNN shows a temporally stable performance, with a good generalization to previously unknown CMLs. The σ_{opt} method performs well only if it is re-calibrated for different months and to individ-

370 ual CMLs, while σ_{q80} is by definition an adaptive method. Even with re-calibration or adaption, the reference methods are outperformed by the CNN.

5 Conclusions

In this study, we explore the performance and robustness of 1D-CNNs for rain event detection in CML attenuation time-series using a large and diverse data set, acquired from 3904 CMLs distributed over entire Germany. We prove that, compared to a

375 reference method, we can minimize the trade-off between false wet and missed wet predictions. While the reference method needs to be adjusted for different months of the analyzed period to provide optimal results, the trained CNN generalizes very well to CMLs and time periods not included in the training data. On average, 87% of all wet and 91% of all dry periods were detected by the CNN, which underlines the strong agreement between rain events that can be detected in the CML time-series and rain events in the RADOLAN-RW data set.

380 In future work, we plan to investigate the potential of using reference data with higher temporal resolution to improve the temporal localization of the rain events. Data with higher temporal resolution will, however, magnify the uncertainties that arise due to the different spatial and temporal coverage of the different rainfall observation techniques. In order to address these uncertainties, it will be important to further explore the relationship between weather radar and CML derived rainfall products. In the study presented here, we focused on the optimization of rain event detection as an isolated processing step,

385 which provides the basis for a successful rain rate estimation. All subsequent processing steps, including WAA correction, k-R relation and spatial interpolation, have an effect on the CML derived rain rate, that can lead to over or under-estimation. Therefore, we plan to study the interplay of different rain event detection methods, including the one presented here, with the different methods of the successive steps of rainfall estimation from CMLs in a larger inter-comparison study.

Our study shows that, using data driven methods like CNNs in combination with the good coverage of the highly developed

390 weather radar network in Germany can lead to robust CML data processing. We anticipate that this robustness enhances the chance that we can transfer processing methods to data from CML networks in developing countries like Burkina Faso, where rainfall information is still scarce despite its high importance to the local population (Gosset et al., 2015).

Code and data availability. Interactive code to build the CNN and an example evaluation using the trained CNN are available at https://github.com/jpolz/cnn_cml_wet-dry_example. CML data was provided by Ericsson Germany and is not publicly available in its full extent.

395 RADOLAN-RW is publicly available through the Climate Data Center of the German Weather Service (DWD) https://opendata.dwd.de/
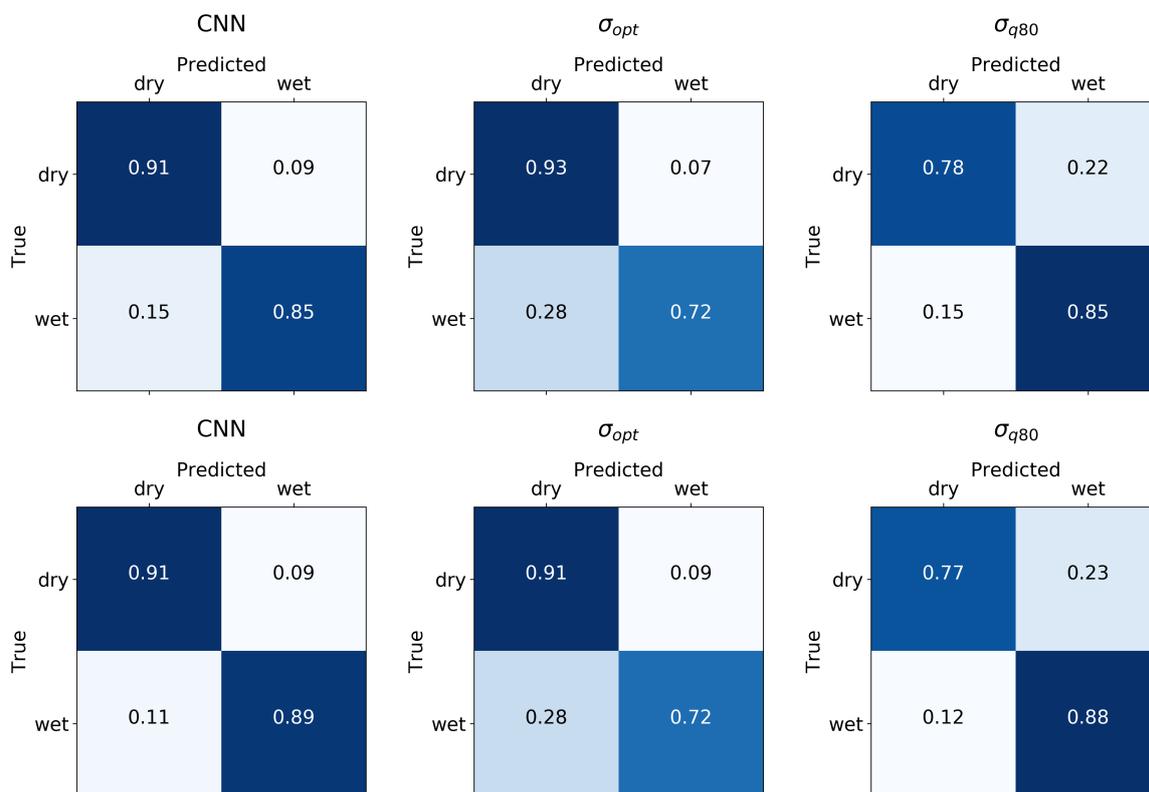
Figure A1. Normalized confusion matrices of VALAPR (top) and VALSEP (bottom).

climate_environment/CDC/grids_germany/hourly/radolan/. We include a small example data set with modified CML locations, the trained model weights and the pre-processed RADOLAN-RW reference data together with the interactive code at https://github.com/jpolz/cnn_cml_wet-dry_example.

Appendix A: Additional Figures

400 **A1**

Author contributions. JP, CC and HK designed the study layout and JP carried it out with contributions of CC and MG. Data was provided by CC and MG. Code was developed by JP with contributions of CC. JP prepared the manuscript with contributions from all co-authors.

Competing interests. The authors declare that they have no conflict of interest.
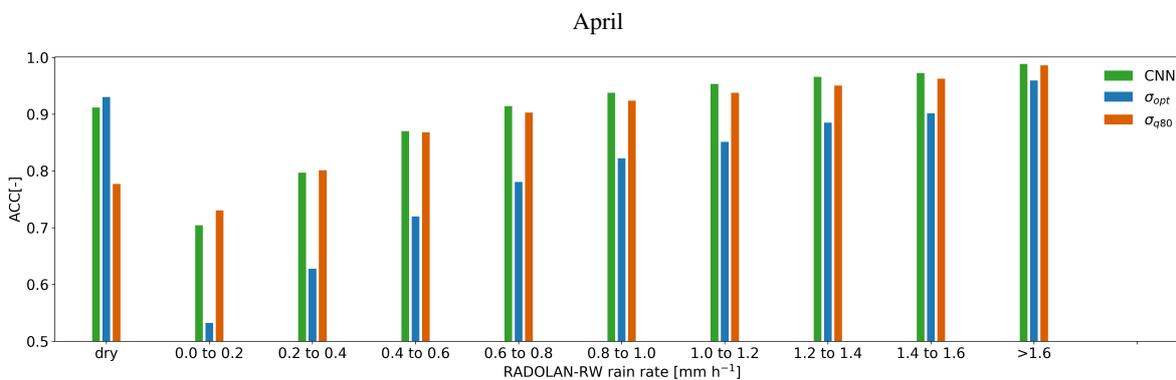
Figure A2. Each bar shows the ACC score on samples from VALAPR, grouped by the reference rain rate.

Acknowledgements. We thank Ericsson for the support and cooperation in the acquisition of the CML data. This work was funded by the
405 German research foundation within the RealPEP research group. Furthermore, we like to thank the German Research Foundation for funding
the project IMAP, the Helmholtz Association of German Research Centres for funding the project Digital Earth and the Bundesministerium
für Bildung und Forschung for funding the project HoWa-innovativ. Special thanks are given to Bumsuk Seo for his valuable advice and for
providing the Titan Xp GPU used for this research, which was donated by the NVIDIA Corporation.

References

Akoglu, H.: User's guide to correlation coefficients, Turkish Journal of Emergency Medicine, 18, 91–93, https://doi.org/10.1016/j.tjem.2018.08.001, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6107969/, 2018.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview, Bioinformatics, 16, 412–424, https://doi.org/10.1093/bioinformatics/16.5.412, https://doi.org/10.1093/bioinformatics/16.5.412, 2000.

Bottou, L., Curtis, F. E., and Nocedal, J.: Optimization Methods for Large-Scale Machine Learning, SIAM Review, 60, 223–311, https://doi.org/10.1137/16M1080173, https://epubs.siam.org/doi/10.1137/16M1080173, 2018.

Bundesnetzagentur: Tätigkeitsbericht Telekommunikation 2016/2017, Tech. rep., Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, Bonn, https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Allgemeines/Bundesnetzagentur/Publikationen/Berichte/2017/TB_Telekommunikation20162017.pdf?__blob=publicationFile&v=3, 2017.

Chollet, F.: Keras, https://github.com/fchollet/keras, 2015.

Chwala, C. and Kunstmann, H.: Commercial microwave link networks for rainfall observation: Assessment of the current status and future challenges, Wiley Interdisciplinary Reviews: Water, 6, e1337, https://doi.org/10.1002/wat2.1337, https://onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1337, 2019.

Chwala, C., Gmeiner, A., Qiu, W., Hipp, S., Nienaber, D., Siart, U., Eibert, T., Pohl, M., Seltmann, J., Fritz, J., and Kunstmann, H.: Precipitation observation using microwave backhaul links in the alpine and pre-alpine region of Southern Germany, Hydrology and Earth System Sciences, 16, 2647–2661, https://doi.org/10.5194/hess-16-2647-2012, https://www.hydrol-earth-syst-sci.net/16/2647/2012/, 2012.

Chwala, C., Keis, F., and Kunstmann, H.: Real-time data acquisition of commercial microwave link networks for hydrometeorological applications, Atmospheric Measurement Techniques, 9, 991–999, https://doi.org/10.5194/amt-9-991-2016, https://www.atmos-meas-tech.net/9/991/2016/, 2016.

Doumounia, A., Gosset, M., Cazenave, F., Kacou, M., and Zougmore, F.: Rainfall monitoring based on microwave links from cellular telecommunication networks: First results from a West African test bed., Geophysical Research Letters, 41, 6016–6022, https://doi.org/10.1002/2014GL060724, http://doi.wiley.com/10.1002/2014GL060724, 2014.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A.: Deep learning for time series classification: a review, Data Mining and Knowledge Discovery, 33, 917–963, https://doi.org/10.1007/s10618-019-00619-1, http://arxiv.org/abs/1809.04356, arXiv: 1809.04356, 2019.

Fawcett, T.: An introduction to ROC analysis, Pattern Recognition Letters, 27, 861–874, https://doi.org/10.1016/j.patrec.2005.10.010, https://www.sciencedirect.com/science/article/abs/pii/S016786550500303X, 2006.

Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, 36, 193–202, https://doi.org/10.1007/BF00344251, https://doi.org/10.1007/BF00344251, 1980.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, The MIT Press, 2016.

Gosset, M., Kunstmann, H., Zougmore, F., Cazenave, F., Leijnse, H., Uijlenhoet, R., Chwala, C., Keis, F., Doumounia, A., Boubacar, B., Kacou, M., Alpert, P., Messer, H., Rieckermann, J., and Hoedjes, J.: Improving Rainfall Measurement in Gauge Poor Regions Thanks to Mobile Telecommunication Networks, Bulletin of the American Meteorological Society, 97, ES49–ES51, https://doi.org/10.1175/BAMS-D-15-00164.1, https://journals.ametsoc.org/doi/full/10.1175/BAMS-D-15-00164.1, 2015.

445   Graf, M., Chwala, C., Polz, J., and Kunstmann, H.: Rainfall estimation from a German-wide commercial microwave link network: Optimized processing and validation for one year of data, Hydrology and Earth System Sciences Discussions, pp. 1–23, https://doi.org/https://doi.org/10.5194/hess-2019-423, https://www.hydrol-earth-syst-sci-discuss.net/hess-2019-423/, 2019.

Habi, H. V. and Messer, H.: Wet-Dry Classification Using LSTM and Commercial Microwave Links, in: 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 149–153, https://doi.org/10.1109/SAM.2018.8448679, 2018.

450   Hoens, T. R. and Chawla, N. V.: Imbalanced Datasets: From Sampling to Classifiers, in: Imbalanced Learning, edited by He, H. and Ma, Y., pp. 43–59, John Wiley & Sons, Inc., Hoboken, NJ, USA, https://doi.org/10.1002/9781118646106.ch3, http://doi.wiley.com/10.1002/9781118646106.ch3, 2013.

Kaufmann, M. and Rieckermann, J.: Identification of dry and rainy periods using telecommunication microwave links., in: 12nd International Conference on Urban Drainage, pp. 10–15, International Water Association, Porto Alegre/Brazil, 2011.

455   Kim, M.-S. and Kwon, B. H.: Rainfall Detection and Rainfall Rate Estimation Using Microwave Attenuation, Atmosphere, 9, 287, https://doi.org/10.3390/atmos9080287, https://www.mdpi.com/2073-4433/9/8/287, 2018.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, Nature, 521, 436–444, https://doi.org/10.1038/nature14539, http://www.nature.com/articles/nature14539, 2015.

Leijnse, H., Uijlenhoet, R., and Stricker, J. N. M.: Rainfall measurement using radio links from cellular communication net-
460   works, Water Resources Research, 43, https://doi.org/10.1029/2006WR005631, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006WR005631, 2007.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster,
465   Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/, 2015.

Messer, H., Zinevich, A., and Alpert, P.: Environmental Monitoring by Wireless Communication Networks, Science, 312, 713–713, https://doi.org/10.1126/science.1120034, https://science.sciencemag.org/content/312/5774/713, 2006.

470   Overeem, A., Leijnse, H., and Uijlenhoet, R.: Measuring urban rainfall using microwave links from commercial cellular communication networks, Water Resources Research, 47, https://doi.org/10.1029/2010WR010350, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010WR010350, 2011.

Overeem, A., Leijnse, H., and Uijlenhoet, R.: Retrieval algorithm for rainfall mapping from microwave links in a cellular communication network, Atmospheric Measurement Techniques, 9, 2425–2444, https://doi.org/10.5194/amt-9-2425-2016, https://www.atmos-meas-tech.
475   net/9/2425/2016/, 2016a.

Overeem, A., Leijnse, H., and Uijlenhoet, R.: Two and a half years of country-wide rainfall maps using radio links from commercial cellular telecommunication networks, Water Resources Research, 52, 8039–8065, https://doi.org/10.1002/2016WR019412, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016WR019412, 2016b.

Pastorek, J., Fencl, M., Rieckermann, J., and Bareš, V.: Commercial microwave links for urban drainage modelling: The ef-
480   fect of link characteristics and their position on runoff simulations, Journal of Environmental Management, 251, 109 522, https://doi.org/https://doi.org/10.1016/j.jenvman.2019.109522, http://www.sciencedirect.com/science/article/pii/S030147971931240X, 2019.

Piczak, K. J.: Environmental sound classification with convolutional neural networks, in: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, IEEE, Boston, MA, USA, https://doi.org/10.1109/MLSP.2015.7324337, http://ieeexplore.ieee.org/document/7324337/, 2015.

Schleiss, M. and Berne, A.: Identification of Dry and Rainy Periods Using Telecommunication Microwave Links, IEEE Geoscience and Remote Sensing Letters, 7, 611–615, https://doi.org/10.1109/LGRS.2010.2043052, 2010.

Smiatek, G., Keis, F., Chwala, C., Fersch, B., and Kunstmann, H.: Potential of commercial microwave link network derived rainfall for river runoff simulations, Environmental Research Letters, 12, 034 026, https://doi.org/10.1088/1748-9326/aa5f46, http://stacks.iop.org/1748-9326/12/i=3/a=034026?key=crossref.bc8f1f4b24b60b13416bb6a85827fcae, 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, 15, 1929–1958, http://jmlr.org/papers/v15/srivastava14a.html, 2014.

Uijlenhoet, R., Overeem, A., and Leijnse, H.: Opportunistic remote sensing of rainfall using microwave links from cellular communication networks, Wiley Interdisciplinary Reviews: Water, 5, e1289, https://doi.org/10.1002/wat2.1289, https://onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1289, 2018.

Wang, Z., Schleiss, M., Jaffrain, J., Berne, A., and Rieckermann, J.: Using Markov switching models to infer dry and rainy periods from telecommunication microwave link signals, Atmospheric Measurement Techniques, 5, 1847–1859, https://doi.org/10.5194/amt-5-1847-2012, https://www.atmos-meas-tech.net/5/1847/2012/, 2012.

Winterrath, T., Rosenow, W., and Weigl, E.: On the DWD quantitative precipitation analysis and nowcasting system for real-time application in German flood risk management, in: Weather Radar and Hydrology, vol. 351, p. 7, IAHS Publ., 2012.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F.: Deep learning in remote sensing: a review, IEEE Geoscience and Remote Sensing Magazine, 5, 8–36, https://doi.org/10.1109/MGRS.2017.2762307, http://arxiv.org/abs/1710.03959, arXiv:1710.03959, 2017.

Đorđević, V., Pronić-Rančić, O., Marinković, Z., Milijić, M., Marković, V., Siart, U., Chwala, C., and Kunstmann, H.: New Method for Detection of Precipitation Based on Artificial Neural Networks, Microwave review, p. 6, 2013.