

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

Julius Polz, Christian Chwala, Maximilian Graf, Harald Kunstmann

Angaben zur Veröffentlichung / Publication details:

Polz, Julius, Christian Chwala, Maximilian Graf, and Harald Kunstmann. 2020. "Rain event detection in commercial microwave link attenuation data using convolutional neural networks." *Atmospheric Measurement Techniques* 13 (7): 3835–53.
<https://doi.org/10.5194/amt-13-3835-2020>.



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

Julius Polz¹, Christian Chwala^{1,2}, Maximilian Graf¹, and Harald Kunstmann^{1,2}

¹Institute of Meteorology and Climate Research, Karlsruhe Institute of Technology, Campus Alpin, Garmisch-Partenkirchen, Germany

²Chair of Regional Climate and Hydrology, Institute of Geography, University of Augsburg, Augsburg, Germany

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

Received: 30 October 2019 – Discussion started: 19 December 2019

Revised: 4 May 2020 – Accepted: 25 May 2020 – Published: 15 July 2020

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 the CMLs, which we treat as a binary classification problem. This processing step is particularly challenging, because even when there is no rain, the signal level can show large fluctuations similar to that during rainy periods. False classifications can have a high impact on falsely estimated rainfall amounts. 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 CNN's ability to recognize attenuation patterns from CMLs and time periods outside the training data. Our CNN is trained on 4 months of data from 800 randomly selected CMLs and validated on 2 different months of data, once for all CMLs and once for the 3104 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 state-of-the-art reference method, which uses the rolling standard deviation of the CML signal level time series 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 76 % of all rainy and 97 % of all nonrainy periods. From all periods with a reference rain rate larger than 0.6 mm h^{-1} , more than 90 % was detected. We also show that the improved event detection leads to a significant reduction of falsely estimated rainfall by up to 51 %. At the same time, the quality of the correctly estimated rainfall is kept at the same level in regards to the Pearson correlation with the radar rainfall. 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 all of Germany.

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 spatiotemporal variability of rainfall, they are a welcome complement to support traditional observations with rain gauges and weather radars, particularly

in regions where radars are 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 by 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., 2020), 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 the 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 all over Germany are recorded continuously with a temporal resolution of 1 min via a real-time data acquisition system (Chwala et al., 2016). The number of existing CMLs across Germany is more than 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 such a big data set. Several studies address the details of the processing steps which are required for deriving rainfall information from CMLs. These steps involve, for example, 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.

1.1 On the importance of rain event detection

The first of these processing steps, called rain event detection, is the separation of rainy (wet) and nonrainy (dry) periods. A static signal-level baseline to derive attenuation that can be attributed to rainfall has proven to be ineffective due to, for example, daily or annual cycles and unexpected jumps in the time series like for CML B in Fig. 1. After the rain events are localized correctly, an event-specific attenuation baseline can be determined and actual rain rates can be de-

rived via the k – R power law which relates specific attenuation k (in dB km^{−1}) to rain rate R (in mm h^{−1}).

Detecting rain events is challenging, because CML signal levels can show high fluctuations, even when there is no rain, e.g., due to multipath propagation (e.g., Chwala and Kunstmann, 2019, Fig. 6). Therefore, the main difficulty is to distinguish between noise and signal fluctuations caused by rain along the CML path. As seen in Fig. 1, the differences in noise levels can vary significantly, depending on the CML that is used. When looking at the magnitude of these fluctuations, we can see that a misclassification of wet and dry periods can easily lead to a large over- or underestimation of rainfall. These missed or falsely estimated quantities are often overlooked in scatter density comparisons of rainfall products like Fig. 9a and b below, which shows our own results. But when absolute amounts are compared, they represent an obvious issue with up to 30 % of the total CML rainfall that can be attributed to false positives. As these misclassifications generate a bias different from the bias corrected in later processing steps like the WAA correction, it is important to optimize the rain event detection as an isolated processing step first and to optimize subsequent processing steps afterwards.

1.2 State of the art

So far, several methods for rain event detection with CMLs have been proposed. The main difference that divides these methods into two groups, is the type of CML data that can be used to estimate rainfall. Depending on the available data acquisition, CML signal levels are either instantaneously sampled at a rate ranging from a few seconds up to 15 min or they are stored as 15 min minimum and maximum values derived from a high instantaneous sampling rate in the background. In almost all cases only one of the two sampling strategies is available due to the type of data management through the network provider. The resulting rain event detection methods are highly optimized for one kind of sampling strategy and therefore in general incompatible with the other kind.

The following methods were developed for instantaneous measurements: Schleiss and Berne (2010) introduced a threshold for the rolling standard deviation (RSD) of the attenuation time series as a criterion to detect rain events. Despite being one of the first methods that were developed, it is still the most commonly used within the CML research community, as it was used in very recent studies from different working groups such as Kim and Kwon (2018), Graf et al. (2020), or Fencel et al. (2020). 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, which was calibrated and validated for a single CML test site. Kaufmann and Rieckermann (2011) have shown the applicability of random forest classifiers and Gaussian factor graphs and validated their approach using 14 CMLs. Dörde-

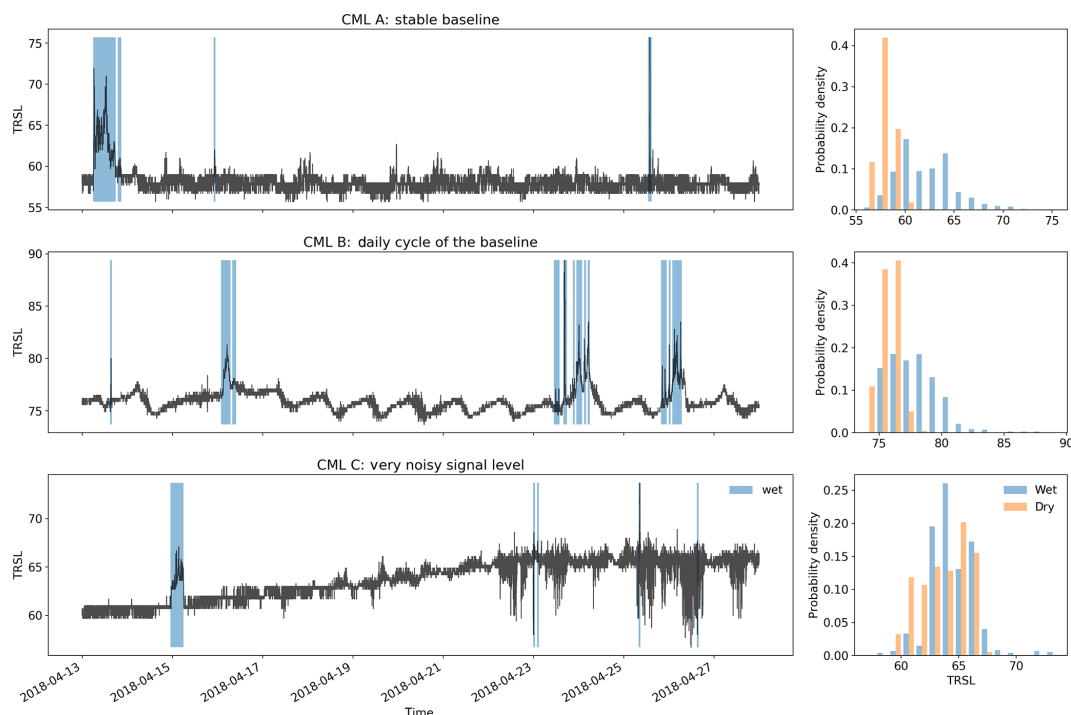


Figure 1. Three example signal level (TRSL) time series that illustrate the high variability in data quality when comparing different CMLs. The blue shaded periods indicate where the radar reference shows rainfall along the CML paths. The challenge is to identify these periods by analyzing the time series. Note that each attenuation event that is falsely classified as wet will produce false rain rate estimates, which will lead to overestimation. The histograms show that for some CMLs the wet periods can be easily separated from the dry periods and for others the distribution of TRSL values is nearly identical for both classes. Figure 2 below will show an example of how different detection methods deal with the challenging time series of CML C.

vić et al. (2013) used a simple multilayer perceptron (MLP) which was trained and validated on a single CML. Ostrometzky and Messer (2018) proposed a simple rolling mean approach to determine a dynamic baseline, also validated on a single CML. Most of these studies are based on a comparably low and sometimes preselected amount of CMLs ranging from 1 to a maximum of 50 devices, a number that is likely much larger in a possible operational setting.

As a detection scheme for 15 min min and max sampled data with a 10 Hz background sampling rate, Overeem et al. (2011) introduced the “nearby link approach”. A period is considered wet if the increase in CML-specific attenuation correlates with the attenuation pattern of nearby CMLs. They concluded that this is only applicable for dense CML networks with a high data availability. Later, they conducted the first evaluation of a rain event detection method on data from 2044 CMLs on a country scale (Overeem et al., 2016b). Very recently the same approach was used in de Vos et al. (2019), showing that this approach works better in combination with min and max sampling than with 15 min instantaneous sampling. Habi and Messer (2018) tested the performance of long short-term memory (LSTM) networks to classify rainy periods from 15 min min and max values of CML signal levels for 34 CMLs.

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

In conclusion, until now, there have been few studies analyzing the performance of rain event detection methods on large data sets. Overeem et al. (2016b) tested the nearby link approach using 2044 CMLs distributed over the Netherlands with a temporal coverage of 2.5 years of data. Graf et al. (2020) extended the RSD method and applied it to 1 year of data from 3904 CMLs to set a benchmark performance on the same data set used in this study. By optimizing thresholds for individual CMLs, the full potential of the RSD method for 1 year of data was explored, yielding good results for the warm season with liquid precipitation. While the RSD method is simple to implement and has only two parameters (window length and threshold) to optimize, it is limited

to measuring the amount of fluctuations rather than the specific pattern. More room for optimization is expected using a data-driven approach, such as machine learning techniques for pattern recognition.

1.3 Data-driven optimization through deep learning

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 learning is also an established method in time series classification (Fawaz et al., 2019). In both studies, 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 (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.

1.4 Research gap and objectives

Due to the opportunistic use of CMLs, the variety of signal fluctuations and possible occurrences of errors naturally increase in a CML data set with its size. Separating rainy from nonrainy periods is therefore a crucial step for rainfall estimation from CMLs. Although applicable on a large scale, recently applied methods still struggle with falsely estimated rainfall as can be seen in the evaluations from Graf et al. (2020) and de Vos et al. (2019). Despite the amount of proposed methods, this processing step has not yet been investigated in detail using a large and diverse CML data set, especially for data-driven approaches. Given their promising results in other applications, the usage of artificial neural networks (ANNs) for rain event detection in the CML attenuation time series on a large scale provides a promising opportunity. It has been proven that in many cases ANNs allow for fast, robust, and high-performance processing of a variety of suitable data sets. What is missing is a proof that they are applicable to a large and diverse CML data set. The question is this: does a high variability of frequency, length, and spatial distribution of the analyzed CMLs or a high variability of rain rates and event duration for a large amount of analyzed periods affect the performance of ANNs in this specific case or not? Additionally, the effect of rain event detection performance on the estimated rain rates has yet to be investigated.

The objective of this study is to evaluate the performance of 1D CNNs to detect rainfall-induced attenuation patterns in

instantaneously measured CML signal levels and to investigate the effect of an improved temporal event localization on the CML-derived rainfall amounts. Furthermore, we test the CNN's ability to transfer its detection performance to new CMLs and future time periods in order to provide a validated open-source model that can be used on other data sets. To provide the CML community with comprehensible results, we compare the CNN to the method of Schleiss and Berne (2010), which we consider to be state of the art due to the amount of recent applications. We aim to provide a high statistical robustness of the derived performance measures by using the, to date, largest available CML data set consisting of data from 3904 CMLs distributed over all of Germany.

2 Methods

The following definition of rain event detection with CMLs is the basis of our methodology: rain event detection is a binary 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.

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 all of 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 sublinks per CML, transmitting their signal in opposite directions along the CML path. For each sublink, a received signal level (RSL) and a transmitted signal level (TSL) are recorded at a temporal resolution of 1 min 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., 2020). The data are available at 97.1 % of all time steps and gaps are mainly due to outages of the data acquisition system.

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 $1 \text{ km} \times 1 \text{ km}$, covering all of Germany on 900×900 grid cells. The temporal resolution is 60 min and the resolution for the rain amount is 0.1 mm (Winterrath et al., 2012). To compare to this refer-

ence, the window length w is set to 60 min 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

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

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 $R_{t,i} \geq 0.1 \text{ mm h}^{-1}$ and dry otherwise.

2.2 Preprocessing

Before training and testing an artificial neural network, the raw time series data have to be preprocessed. We do this to sample time windows of a fixed size, which are normalized and labeled 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 800 randomly chosen CMLs in the period from May to August 2018. VALAPR covers the remaining 3104 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 20 % of all available CMLs for training allows us to analyze the CNN's generalization of the remaining CMLs in the validation data set. No CMLs were excluded from this analysis.

For each of the two sublinks of a CML, we compute a transmitted minus received signal level (TRSL). Within one TRSL time series, randomly occurring gaps of up to 5 min of missing data are linearly interpolated to be consistent with the preprocessing used in Graf et al. (2020). 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 the model performance. We perform the normalization as a preprocessing 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 all available data from the preceding 72 h from each time step. In rare cases of larger gaps in the data acquisition, we set a lower limit for the data availability to 120 min.

The set of starting timestamps of the hourly reference data set is denoted T_{rad} . For each CML i and each starting time $t \in T_{\text{rad}}$, a sample of data $\bar{X}_{t,i}$ is composed from 60 + k minutes of TRSL from the two sublinks starting at $t - k$. The first k minutes serve as a reference to previous behavior of the same CML and the last 60 min are the period $X_{t,i}$ that has to be classified. To investigate the impact of adding this additional information, we compare multiple setups with k ranging from 0 to 240 min. The results are given in Sect. 3. An example TRSL over a period of 2 weeks is shown in Fig. 2a.

After interpolating short gaps, as described above, we exclude all samples with missing values from the analysis. Since we lose up to 5 h of data whenever there is a gap, the interpolation routine increases the number of available samples from 75 % to 94 %.

To train the CNN, we have to balance the wet and dry classes in the data set (Hoens and Chawla, 2013). The undersampling 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 % and 10 %. 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 original unbalanced training data set is not used in the analysis. In total, the numbers of samples are 2.3×10^5 for TRG, 3.9×10^5 for VALAPRB, 2.2×10^6 for VALAPR, and 2.8×10^6 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. 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, CNNs consist of a series of layers of neurons (Fig. 3). 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.

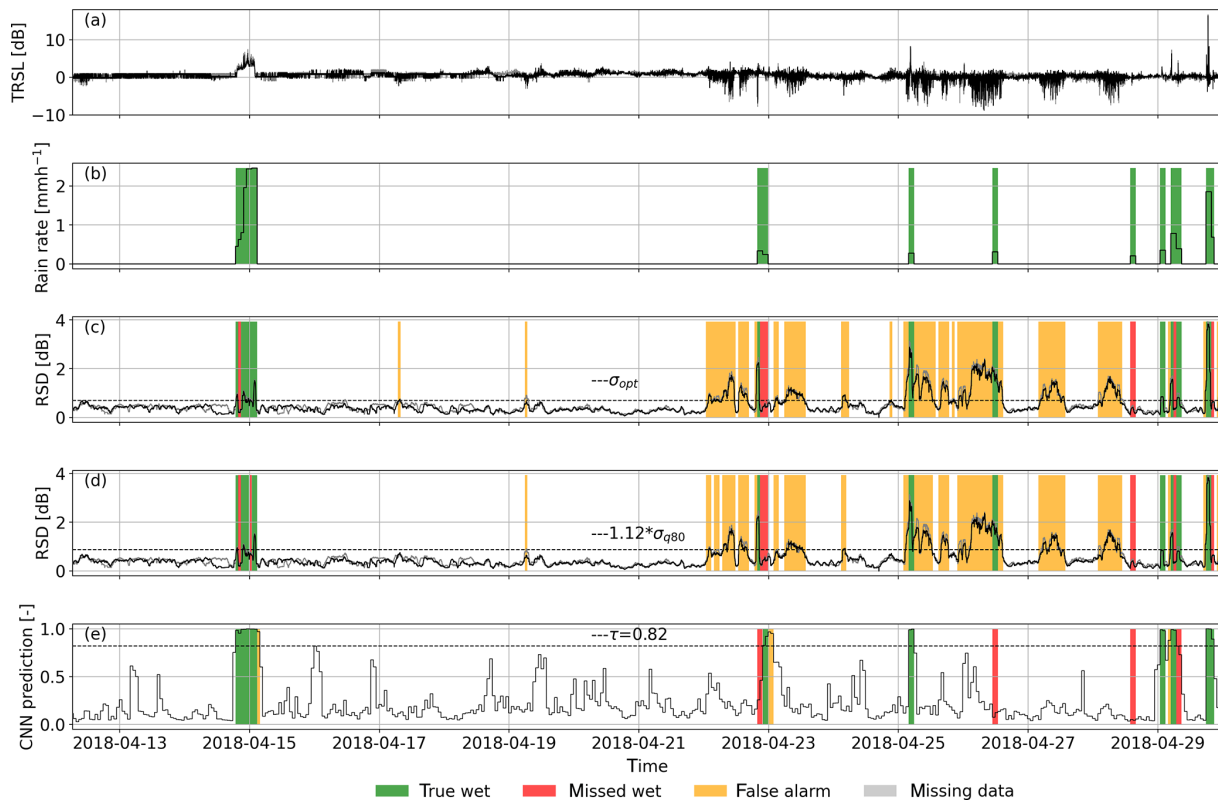


Figure 2. Performance of the CNN and the reference methods for the noisy example CML time series from Fig. 1. Panel (a) shows the normalized TRSL time series and (b) is the radar reference. Predictions from the CNN (e) yield a Matthews correlation coefficient (MCC) of 0.74. Predictions through σ_{opt} (c) and σ_{q80} (d), which are very similar in this case, both yield MCCs of 0.28. Note that the TRSL and RSD time series of sublink 2 are almost identical to those of sublink 1 and are shown in light gray.

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 of the convolutional layers is one-dimensional data. 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. 3). Convolutional layers extract feature maps

by passing local patches (3×1) 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 (respectively 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 overfitting to the training data, two dropout layers are added, one after each fully connected layer, with a dropout ratio of 0.4 (Srivastava et al., 2014).

We implement the CNN in a Python framework using the Keras (version 2.3.1) back end for TensorFlow (version 2.1.0) (Chollet, 2015; Abadi et al., 2015). For the model architecture, the type, number, and order of layers have 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

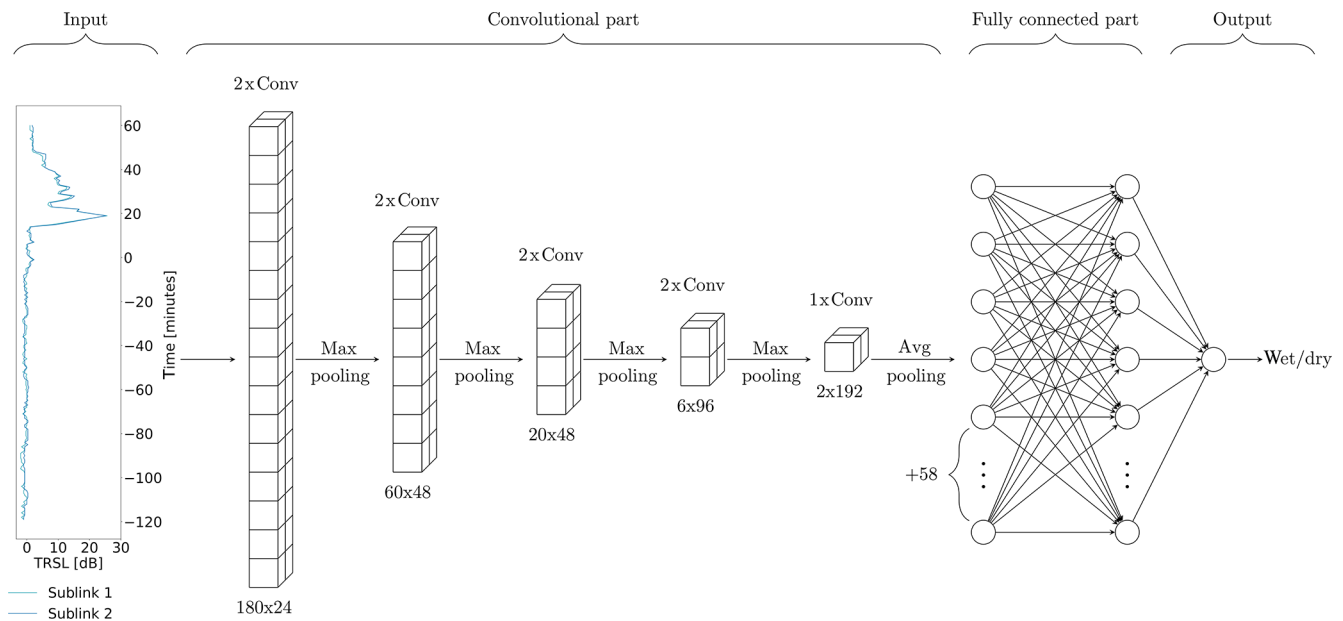


Figure 3. Graphical illustration of the CNN's architecture for $k = 120$. The input shows one sample $\bar{X}_{t,i}$ of data consisting of 180 min of TRSL from the two sublinks of one CML. Convolutional and pooling layers reduce the input dimension from 180 to 2, while a total of 192 features are extracted. 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.

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 Sect. 2.4). Depending on the length of the input time series, which varies with k , the number of convolutional layers is different, i.e., when $k < 60$ we omit the last two convolution layers. We trained one model for each value of k and one extra model that additionally receives the CML metadata consisting of the length and the frequency of both channels through parallel fully connected layers and an add layer before the fully connected part. For k set to 120 min, 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. 3. For all model versions, the detailed model and training specifications, all hyper parameters, and the weights of the trained CNN can be retrieved from the code example (Polz, 2020).

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 a stochastic gradient descent with binary cross-entropy as a loss function and an initial learning rate of 0.008 (Bot-

to 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 to compute the training and validation loss, and the learning rate is decreased slightly.

The training is stopped if the validation loss does not equal or surpass an earlier minimal value for 50 epochs (stopping criterion). Afterwards the model which achieves the best validation Matthews correlation coefficient (see MCC below) is selected from all versions that existed after the individual training epochs (model selection criterion). This model is then used for classification on the validation data sets.

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. 5), 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

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

Classification results, in the form of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) 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

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

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad (4)$$

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

and

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

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

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

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.

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

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \in [-1, 1], \quad (8)$$

where an MCC of 0 represents random guessing and an MCC of 1 represents a perfect classification. A strong correlation is given at values above 0.25 (Akoglu, 2018). The advantage of the MCC is that it is a single number which we use to optimize the threshold for the CNN.

The third metric we use is the receiver operating characteristic (ROC), defined by the pair $(\text{FPR}, \text{TPR}) \in [0, 1] \times [0, 1]$ (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 therefore a climatologically independent measure for the classifier's performance on rain event detection. Each $\tau \in [0, 1]$ leads to 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

$$\text{AUC} = \int_0^1 \gamma d\tau \in [0, 1] \quad (9)$$

under the ROC curve (AUC). 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 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 . The main purpose of the ROC is that we use it to compare different methods, e.g., different values of k , independent from a fixed threshold, by considering the ROC curve and the AUC.

2.5 Reference method

The reference method is a modification of Schleiss and Berne (2010), which is to date the most commonly used method to separate wet and dry periods as reviewed in the introduction. 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 fluctuations, and there are multiple effects that can cause high signal fluctuations during dry periods, e.g., for CML C in Fig. 1. Some of the factors are known, like multipath 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 min 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 sublinks exceeds 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 min rolling standard deviation of 1 month for a certain CML multiplied by a scaling factor 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. (2020). The second one, denoted σ_{opt} , is optimized against the reference 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.

2.6 Rain rate estimation

In the same way as the rolling standard deviation, the CNN can be used in a rolling window approach, classifying the

Table 1. Confusion matrix.

Prediction	Reference		
	Wet		Dry
	Wet	True wet (TP): #{ detected wet reference wet}	False wet (FP): #{ detected wet reference dry}
	Dry	Missed wet (FN): #{ detected dry reference wet}	True dry (TN): #{ detected dry reference dry}

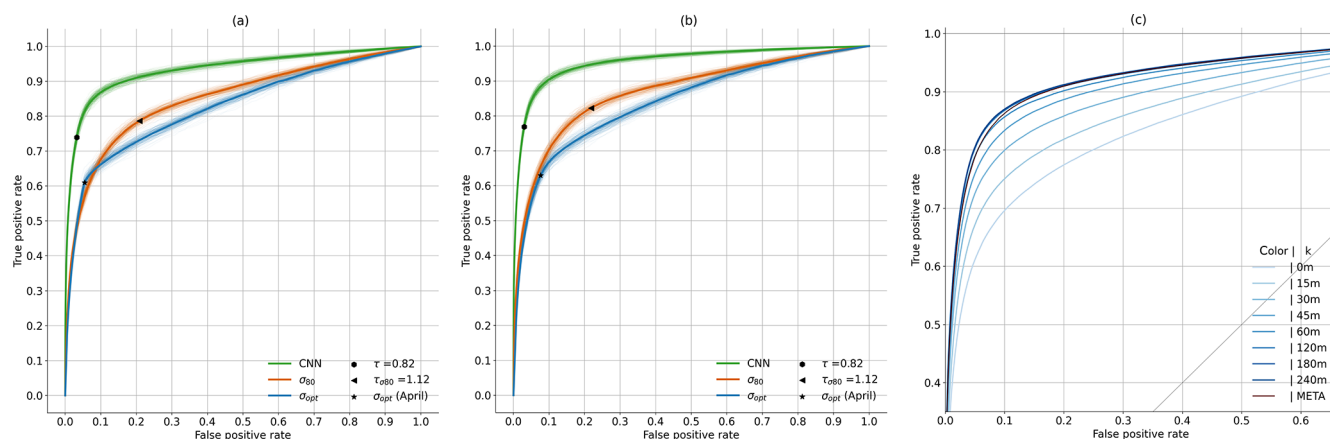


Figure 4. Receiver operating characteristic (ROC) curves on VALAPR (a) and VALSEP (b). Fine lines are generated by 200 random selections (bootstrapping) of 1 % of the samples and account for the variability of the model performance during a random short period (~ 8 h) of data. The performances of the CNN for different values of k and the added metadata are shown in (c), and the AUC values are given in Table B1.

timestamp t as wet or dry by using the sample with starting timestamp $t - 30$ as model input. With the resulting rain event detection information from either the CNN or the two reference methods, rain rates are estimated in several steps. We use the exact same processing scheme as described in Graf et al. (2020), which we refer the reader to for all the technical details. This processing includes erratic treatment of CMLs and WAA compensation to derive rain rates with a temporal resolution of 1 min. For each detected rain event, a constant baseline of the TRSL is calculated from the preceding dry period. The attenuation above this baseline level is attributed to rain but also to WAA. The WAA is compensated depending on the rain rate, using a method modified after Leijnse et al. (2008). The remaining specific attenuation k is used to derive the path-averaged rain rate R using the $k - R$ relation from Eq. (10). The constants a and b are taken from ITU (2005).

$$k = aR^b \quad (10)$$

For the CMLs used in this study, this relation is close to linear, i.e., b is close to one. For a comparison to RADOLAN-RW, the 1 min rain rates are then aggregated by taking the hourly average.

Only from this analysis, data from 45 CMLs (1.1 %) are discarded due to substantially erratic signal levels to be able to follow the same procedure as in Graf et al. (2020). Additionally, we justify this procedure with the following obser-

vation: for the rain event detection, we want periods of erratic behavior to be included in both training and validation data, since also CMLs that are not discarded by the erratic treatment can show periods of erratic behavior, such as CML C from Fig. 1. Each erratic training and validation sample contributes to the final statistics as one sample and the erratic CMLs do not distort the analysis. This is very different for the rainfall amount, since erratic links are prone to a very high overestimation of the final rain rates even when a low number of time periods are detected wet. Since erratic CMLs are a small fraction of the available CMLs and they can be detected automatically, we decided to exclude their bias when analyzing the contribution of false positives to absolute rainfall amounts. An example of such a time series can be found in Fig. A2.

3 Results

During training on TRG, the performance of the CNN was evaluated on VALAPRB after each epoch. The resulting graphs of loss, ACC, TPR, and TNR during the training process are shown in Fig. 6. 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 VALAPR, by maximizing the MCC, with resulting values of τ shown in Table B1. The

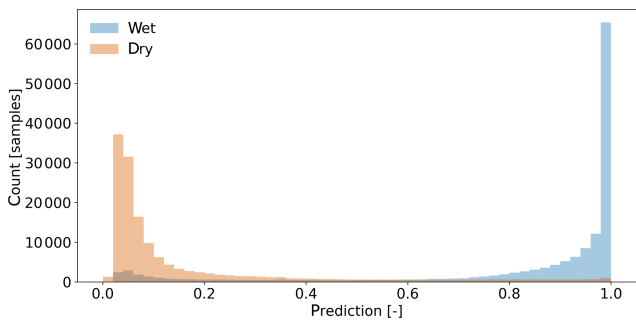


Figure 5. Raw CNN predictions on VALAPRB, colored according to the reference.

results from that table and the ROC curves in Fig. 4c show that in general the performance of the CNN is increasing with higher values of k , but the performance gain was insignificant for raising the value higher than 120 min or adding metadata as model input. We therefore decided to set $k = 120$ and not to use added metadata for evaluating further results and comparing them to the reference methods.

Figure 5 shows the distribution of the CNN’s predictions on VALAPRB. The threshold τ is set to 0.82. The final number of training epochs was 248, and the model from epoch 212 was selected (see Fig. 6a). On one Nvidia Titan Xp GPU the training time was 30 min. Classifying 3904 samples, i.e., a 1 min time step for all CMLs, took 20 ms, which can be considered extremely fast, allowing for a real-time application of the method. For further verification, we repeated the training multiple times with a different randomization (selection of CMLs and balancing) of TRG and VALAPRB, but no significant changes in performance could be observed.

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 TPR values of 0.74 (VALAPR) and 0.77 (VALSEP) and TNR values of 0.97 (VALAPR and VALSEP) (see also Fig. A1). On average, only 3 % of the dry periods was falsely classified as wet and 24 % of the wet periods was missed. With a scaling factor $\tau_{\sigma_{q80}}$ of 1.12, σ_{q80} achieved a balanced TPR and TNR with a value of around 0.79 for both rates in April and September. σ_{opt} , on the other hand, achieved similar TNR values than the CNN but at the cost of lower TPR values.

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 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 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

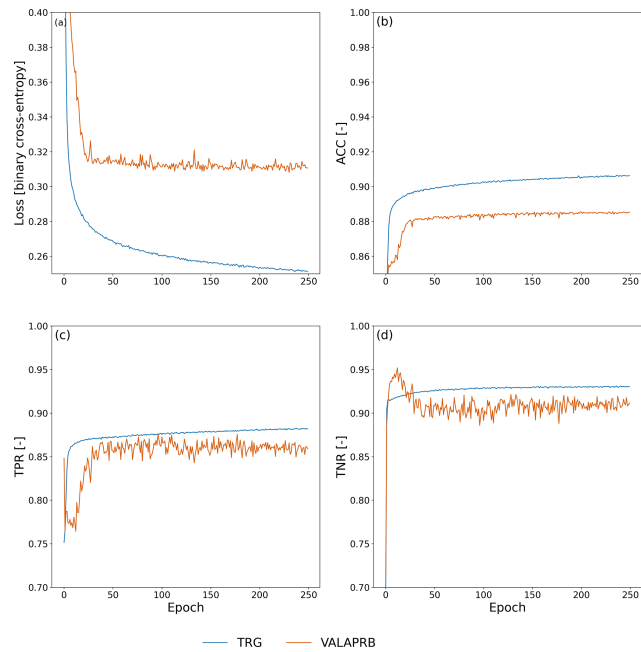


Figure 6. Statistics of variables that were monitored during the training process.

Table 2. Performance metrics of rain event detection methods on VALAPR and VALSEP. Best performances for the individual metrics are highlighted (bold).

	Method	TPR	TNR	ACC	MCC	AUC
VALAPR	CNN	0.74	0.97	0.95	0.69	0.94
	σ_{q80}	0.79	0.79	0.79	0.38	0.85
	σ_{opt}	0.61	0.95	0.91	0.52	0.83
VALSEP	CNN	0.77	0.97	0.96	0.69	0.96
	σ_{q80}	0.82	0.78	0.78	0.35	0.87
	σ_{opt}	0.63	0.92	0.90	0.44	0.84

of 0.95 and 0.69 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. 7. From all rain events where $R_{t,i} \geq 0.6$ mm, 90.4 % was correctly detected by the CNN. On the other hand, around 38.9 % of all rain events with $R_{t,i} < 0.6$ mm was 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 or σ_{opt} .

The MCC was computed individually for each CML and each validation data set. Figure 8 shows scatter density plots comparing the individual MCC scores of the CNN and σ_{opt} .

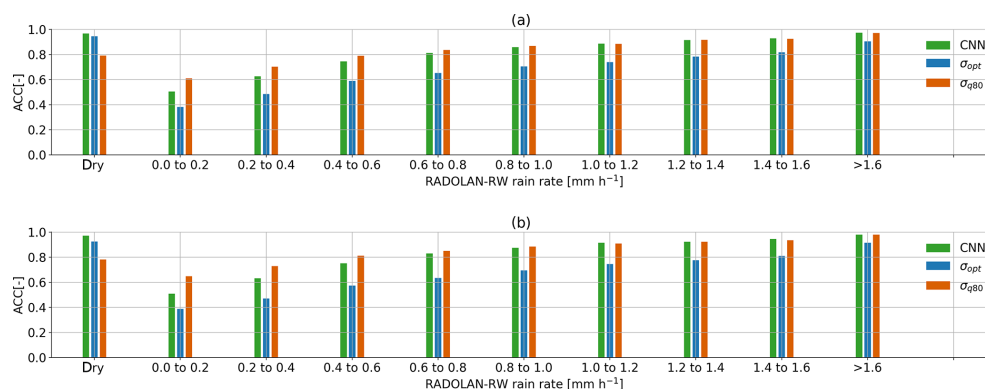


Figure 7. Each bar shows the ACC score on samples from (a) VALAPR and (b) VALSEP, grouped by the reference rain rate. An ACC of 0.5 represents random guessing.

The CNN's MCC for VALAPR is higher for 95.9 % of all CMLs, and for VALSEP it is higher for 96.7 % of all CMLs.

We focus our analysis on hourly rainfall rates from all non-erratic CMLs in September 2018. The resulting rain rates using either the CNN or the σ_{q80} detection scheme are shown in Fig. 9. For both methods, the distribution of false positive and false negative samples is centered around 0.1 mm h^{-1} , and the distribution of true positives is centered around 1 mm h^{-1} . While the percentage of CML-derived rainfall estimated during false positive events is 29.9 % for σ_{q80} , it is significantly less for the CNN (see Fig. 9d and f). This constitutes a reduction of 51 % of falsely estimated rainfall for the month of September 2018. At the same time the amount of missed rainfall is reduced by 27.5 %. The amount of rainfall in the true positive category could therefore be raised by 4.7 %. The Pearson correlation for the hourly rainfall estimates between radar and CMLs is 0.83 using σ_{q80} and 0.84 using the CNN.

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 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 overall performance is depending on the distribution of the intensity of rain events (see Fig. 7) and since there is a large variability of performance between the CMLs (see Fig. 8).

Since the results on both validation data sets are very similar (see Table 2), we further focus on VALSEP, which was not used to optimize the model hyper-parameters. With an ACC of 0.95 and an MCC of 0.69, the correlation between the CNN predictions and the reference data set RADOLAN-

RW can be considered very high. A TPR of 0.74 might not appear very good at first sight, but considering that the detection accuracy for samples with a rain rate of smaller than 0.6 mm h^{-1} is only 0.61, we actually achieve an accuracy of over 0.9 for all rain rates higher than 0.6 mm h^{-1} .

The CNN and the reference method σ_{opt} have a similar ACC value. At the same time the CNN's MCC is 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 0.13. This constitutes a major improvement by the CNN. As shown in Fig. 8, 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 distribution of MCC values of individual CMLs is the same in April and September, while performance drops for σ_{opt} . The CNN's improvement in ACC and MCC over σ_{q80} was even higher with 0.17 and 0.32. While the TPR of σ_{q80} is slightly higher than the TPR of the CNN, the TNR is much lower for σ_{q80} . Thus the CNN shows substantial improvement in correctly classifying dry periods.

While the RSD method can be set up to either have a high TPR (σ_{q80}) or a high TNR (σ_{opt}), the ROC curves show that 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 observation is illustrated by the example in Fig. 2, which shows a very noisy CML time series that produces a high amount of false positives for the reference method, while the CNN does not attribute these fluctuations to rainfall.

All three methods have limitations to detect events with rain rates smaller than 0.3 mm . 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

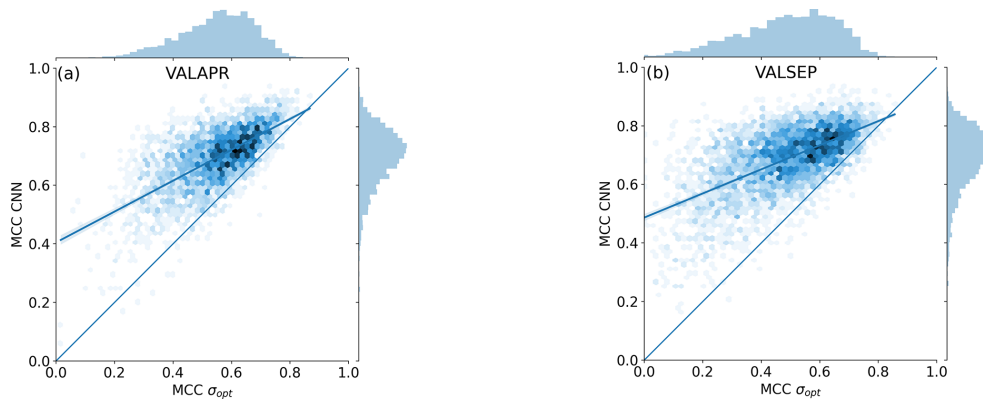


Figure 8. Scatter density plots of the MCC achieved by the CNN and σ_{opt} on data from individual CMLs. Both methods are MCC optimized for the unbalanced data from VALAPR; while the CNN keeps the optimized performance in September, the performance of σ_{opt} drops.

< 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 (0.3 dB for RSL and 1 dB for TSL) might therefore not allow for a detectable signal.

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 is in the critical range from 0.1 to 0.3 mm, there is 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.

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 is 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 is, according to the CNN, very likely to belong to the falsely predicted class. Despite this being an issue for many CMLs, about 10 % has 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 for future applications in quantitative precipitation estimation with CMLs.

4.2 Robustness

The CNN's ability to transfer the detection performance to generalize 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. 6). As expected, the training data showed better performance, but the validation was close at all epochs.

Only 20 % of all available CMLs was used for training. The remaining 80 % was only used to prevent the model from over-fitting to the training data, to choose the model architecture, and to optimize the single parameter τ . Thus no information about the validation data was given directly to the model. The resulting model architecture and hyperparameters 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. 6 therefore prove that the CNN was able to recognize the attenuation pattern in the signal levels of a large number of previously unknown CMLs.

The stability of the CNN's 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. 7a). When we compare the CNN's 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 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 ran-

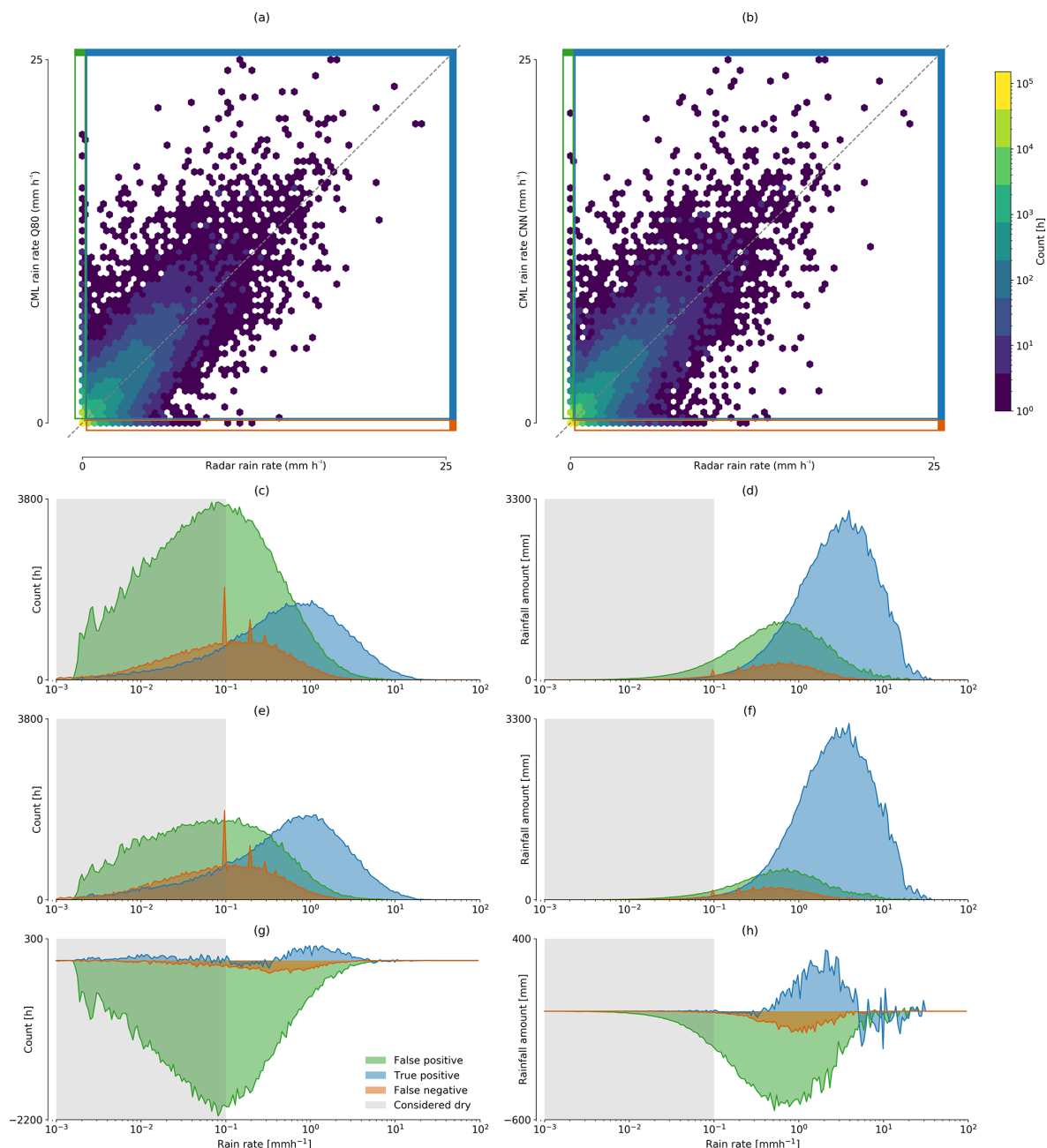


Figure 9. Scatter density comparison between hourly CML and radar rain rate estimates derived from (a) σ_{q80} and (b) the CNN. On the left-hand side the amount of FP, TP, and FN hours with a specific rain rate are compared for (c) σ_{q80} , (e) the CNN, and (g) their difference. On the right-hand side the amounts of rainfall these hours contribute are shown for (d) σ_{q80} , (f) the CNN, and (h) their difference. The rain rates for false positives and true positives are estimated by the CML, while the rain rates for false negatives are taken from the reference.

dom subsets of the validation data. The CNN shows the lowest variability.

As a measure for the flexibility of a classifier, we adopted the ROC analysis in Sect. 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, 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 of previously unknown CMLs. The σ_{opt} method performs well only if it is recalibrated for different months and to individual CMLs, while σ_{q80} is by definition an adaptive method. Even with recalibration or adaption, the reference methods are outperformed by the CNN.

4.3 Impact of the detection scheme on the derived rainfall amounts

The difference between the scatter density plots in Fig. 9a and b seems to be quite low at first sight. What this representation of the data is not stressing enough is the amount of rainfall generated by false positives. But they are an issue that is clearly visible from Fig. 9c–h. Considering that the amount of rainfall estimated during time periods falsely classified as wet can be reduced by 51.0 % and that the amount of rainfall from missed events can be reduced by 27.4 %, the CNN shows a major improvement over the reference method. The 4.1 % of additional rainfall in the correctly classified wet periods stems from time periods that were originally harder to classify, i.e., from small rain events, and it should be expected that the correlation between CML and radar rainfall drops. Instead, the Pearson correlation coefficient increased slightly, showing that the quality of the estimated hourly rainfall could be improved. We omitted the same analysis for a comparison of the CNN and σ_{opt} for which, based on the ROC values in Fig. 4, we anticipate a similar result but with a higher pronunciation of missed rain events instead of the strong impact of false positives.

Overall, we could observe that the improvement in rain event detection has a considerable effect on the amount of over- or underestimation through falsely detected or missed rain events. The improvement in the trade-off between false positives and false negatives directly translates to the impact of their respective rainfall amounts. This is shown by the false positive and false negative distributions in Fig. 9c–f, which are centered around the same value, but are different in their amount depending on the used detection method.

5 Conclusions

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 all of Germany. We prove that, compared to a 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 shows a stable performance for CMLs and time periods not included in the training data very well. On average, 76 % of all wet, and 97 % of all dry, periods was detected by the CNN.

For rain rates higher than 0.6 mm h^{-1} , more than 90 % was correctly detected. This 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.

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, 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 also lead to over or underestimation. While 29.9 % of the estimated rainfall through the reference method can be attributed to false positive classifications, the CNN reduces this amount by up to 51 % and, at the same time, improves on true positive and false negatives. We anticipate that this improvement will lead to new insights into other effects that may disturb the quality of this opportunistic sensing approach.

Our study shows that using data-driven methods like CNNs in combination with the good coverage of the highly developed 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 other CML networks, particularly in developing countries like Burkina Faso, where rainfall information is still scarce despite its high importance to the local population (Gosset et al., 2016).

Appendix A: Additional figures

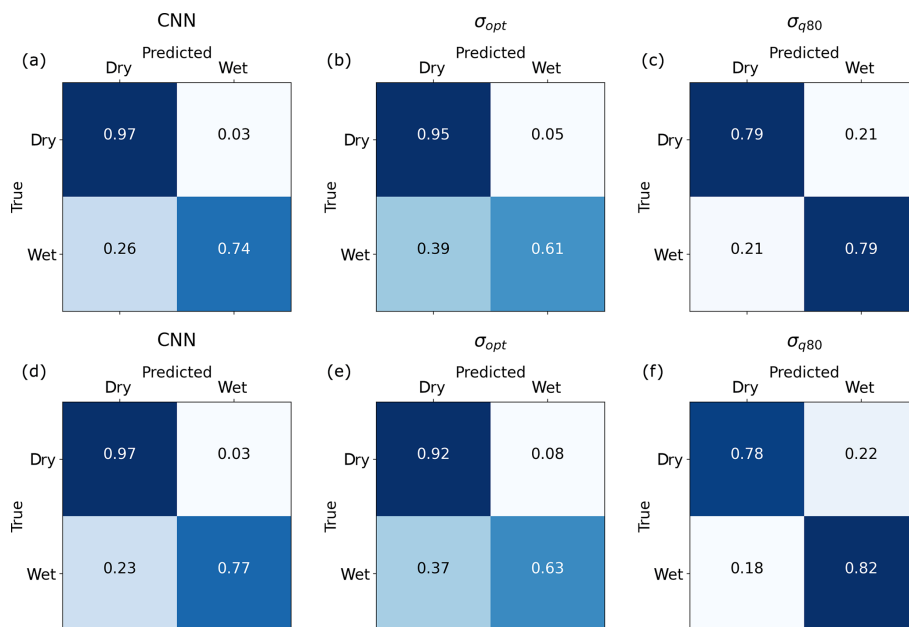


Figure A1. Normalized confusion matrices of VALAPR (a, b, c) and VALSEP (d, e, f).

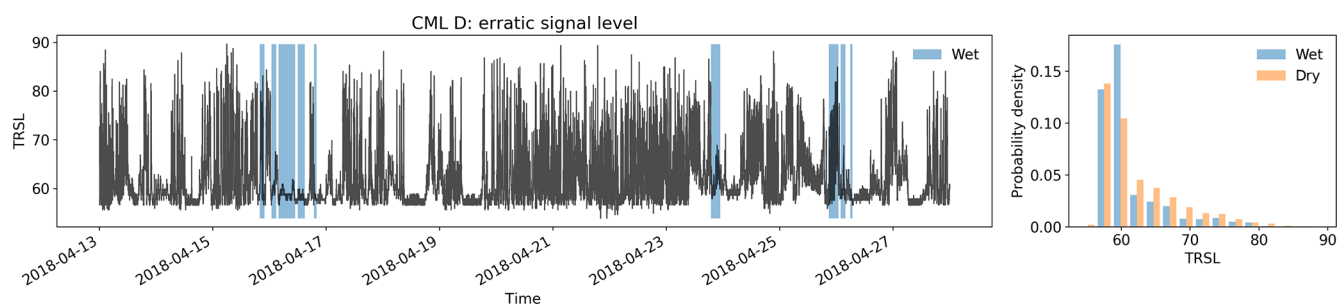


Figure A2. Time series of a CML that is considered erratic and is removed by the simple filter for erratic CML data introduced in Graf et al. (2020). There are no time periods where a reasonable rainfall estimation would be possible.

Appendix B: Additional table

Table B1. Number of training epochs, MCC optimized threshold, and resulting metrics for different values of k evaluated on VALAPR.

Method	k	Training epochs	Threshold τ	TPR	TNR	ACC	MCC	AUC
CNN	0	269	0.77	0.53	0.97	0.93	0.55	0.86
	15	158	0.78	0.59	0.97	0.94	0.60	0.88
	30	274	0.79	0.64	0.97	0.94	0.64	0.91
	45	271	0.79	0.67	0.97	0.94	0.66	0.92
	60	128	0.84	0.71	0.97	0.95	0.68	0.93
	120	212	0.85	0.72	0.97	0.95	0.69	0.94
	180	211	0.86	0.72	0.97	0.95	0.69	0.94
	240	170	0.84	0.73	0.97	0.95	0.69	0.94
CNN+meta	180	321	0.79	0.70	0.97	0.95	0.68	0.93
σ_{q80}	–	–	–	0.79	0.79	0.79	0.38	0.85
σ_{opt}	–	–	–	0.61	0.95	0.91	0.51	0.83

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 (last access: 2 July 2020; Polz, 2020). CML data were provided by Ericsson Germany and are not publicly available to their full extent. RADOLAN-RW is publicly available through the Climate Data Center of the German Weather Service (DWD): https://opendata.dwd.de/climate_environment/CDC/grids_germany/hourly/radolan/ (last access: 2 July 2020; DWD CDC, 2020). We include a small example data set with modified CML locations, the trained model weights, and the preprocessed RADOLAN-RW reference data together with the interactive code (Polz, 2020).

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

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank Ericsson, especially Reinhard Gerigk, Michael Wahl, and Declan Forde for their support and co-operation in the acquisition of the CML data. This work was funded by the German Research Foundation within the RealPEP research group. Furthermore, we like to thank the German Research Foundation for funding the project IMAF, 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.

Financial support. This research has been supported by the Deutsche Forschungsgemeinschaft (grant nos. CH 1785/1-1 and KU 2090/7-2), the Bundesministerium für Bildung und Forschung (grant no. 13N14826), and the Helmholtz Association (grant no. ZT-0025).

The article processing charges for this open-access publication were covered by a Research Centre of the Helmholtz Association.

Review statement. This paper was edited by Gianfranco Vulpiani and reviewed by Andreas Scheidegger and one anonymous referee.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Joze-fowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, Olah, D., Schuster, C., Shlens, M., Steiner, J., Sutskever, B., Talwar, I., Tucker, K., Vanhoucke, P., Vasudevan, V., Viégas, V., Vinyals, F., Warden, O., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, available at: <https://www.tensorflow.org/> (last access: 2 July 2020), 2015.
- 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>, 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>, 2000.
- Bottou, L., Curtis, F. E., and Nocedal, J.: Optimization Methods for Large-Scale Machine Learning, *SIAM Rev.*, 60, 223–311, <https://doi.org/10.1137/16M1080173>, 2018.
- Bundesnetzagentur: Tätigkeitsbericht Telekommunikation 2016/2017, Tech. rep., Report 2016/2017, Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, Bonn, available at: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Allgemeines/Bundesnetzagentur/Publikationen/Berichte/2017/TB_Telekommunikation20162017.pdf?__blob=publicationFile&v=3 (last access: 2 July 2020), 2017.
- Chollet, F.: Keras, GitHub, available at: <https://github.com/fchollet/keras> (last access: 2 July 2020), 2015.
- Chwala, C. and Kunstmann, H.: Commercial microwave link networks for rainfall observation: Assessment of the current status and future challenges, *WIREs Water*, 6, e1337, <https://doi.org/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, *Hydrol. Earth Syst. Sci.*, 16, 2647–2661, <https://doi.org/10.5194/hess-16-2647-2012>, 2012.
- Chwala, C., Keis, F., and Kunstmann, H.: Real-time data acquisition of commercial microwave link networks for hydrometeorological applications, *Atmos. Meas. Tech.*, 9, 991–999, <https://doi.org/10.5194/amt-9-991-2016>, 2016.
- de Vos, L. W., Overeem, A., Leijnse, H., and Uijlenhoet, R.: Rainfall Estimation Accuracy of a Nationwide Instantaneously Sampling Commercial Microwave Link Network: Error Dependency on Known Characteristics, *J. Atmos. Ocean. Tech.*, 36, 1267–1283, <https://doi.org/10.1175/JTECH-D-18-0197.1>, 2019.
- Dorđević, 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*, 19/2, 50–55, 2013.
- 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, *Geophys. Res. Lett.*, 41, 6016–6022, <https://doi.org/10.1002/2014GL060724>, 2014.

- DWD CDC (Deutscher Wetterdienst Climate Data Center): RADOLAN-RW, available at: https://opendata.dwd.de/climate_environment/CDC/grids_germany/hourly/radolan/, last access: 2 July 2020.
- Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A.: Deep learning for time series classification: a review, *Data Min. Knowl. Disc.*, 33, 917–963, <https://doi.org/10.1007/s10618-019-00619-1>, 2019.
- Fawcett, T.: An introduction to ROC analysis, *Pattern Recogn. Lett.*, 27, 861–874, <https://doi.org/10.1016/j.patrec.2005.10.010>, 2006.
- Fencl, M., Dohnal, M., Valtr, P., Grabner, M., and Bareš, V.: Atmospheric observations with E-band microwave links – challenges and opportunities, *Atmos. Meas. Tech. Discuss.*, <https://doi.org/10.5194/amt-2020-28>, in review, 2020.
- Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.*, 36, 193–202, <https://doi.org/10.1007/BF00344251>, 1980.
- Goodfellow, I., Bengio, Y., and Courville, A.: *Deep Learning*, The MIT Press, Cambridge, Massachusetts, 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, *B. Am. Meteorol. Society*, 97, ES49–ES51, <https://doi.org/10.1175/BAMS-D-15-00164.1>, 2016.
- Graf, M., Chwala, C., Polz, J., and Kunstmann, H.: Rainfall estimation from a German-wide commercial microwave link network: optimized processing and validation for 1 year of data, *Hydrol. Earth Syst. Sci.*, 24, 2931–2950, <https://doi.org/10.5194/hess-24-2931-2020>, 2020.
- 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), 149–153, <https://doi.org/10.1109/SAM.2018.8448679>, 2018.
- Hoens, T. R. and Chawla, N. V.: Imbalanced Datasets: From Sampling to Classifiers, in: *Imbalanced Learning*, edited by: He, H. and Ma, Y., John Wiley & Sons, Inc., Hoboken, NJ, USA, 43–59, <https://doi.org/10.1002/9781118646106.ch3>, 2013.
- ITU: ITU-R: Specific attenuation model for rain for use in prediction methods, Tech. Rep. (Recommendation P.838-3), ITU, Geneva, Switzerland, available at: <https://www.itu.int/rec/R-REC-P.838-3-200503-I/en> (last access: 2 July 2020), 2005.
- Kaufmann, M. and Rieckermann, J.: Identification of dry and rainy periods using telecommunication microwave links, in: 12nd International Conference on Urban Drainage, International Water Association, Porto Alegre, Brazil, 10–15 September 2011.
- 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>, 2018.
- LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, *Nature*, 521, 436–444, <https://doi.org/10.1038/nature14539>, 2015.
- Leijnse, H., Uijlenhoet, R., and Stricker, J. N. M.: Rainfall measurement using radio links from cellular communication networks, *Water Resour. Res.*, 43, W03201, <https://doi.org/10.1029/2006WR005631>, 2007.
- Leijnse, H., Uijlenhoet, R., and Stricker, J. N. M.: Microwave link rainfall estimation: Effects of link length and frequency, temporal sampling, power resolution, and wet antenna attenuation, *Adv. Water Resour.*, 31, 1481–1493, <https://doi.org/10.1016/j.advwatres.2008.03.004>, 2008.
- Messer, H., Zinevich, A., and Alpert, P.: Environmental Monitoring by Wireless Communication Networks, *Science*, 312, 713–713, <https://doi.org/10.1126/science.1120034>, 2006.
- Ostrometzky, J. and Messer, H.: Dynamic Determination of the Baseline Level in Microwave Links for Rain Monitoring From Minimum Attenuation Values, *IEEE J. Sel. Top. Appl.*, 11, 24–33, <https://doi.org/10.1109/JSTARS.2017.2752902>, 2018.
- Overeem, A., Leijnse, H., and Uijlenhoet, R.: Measuring urban rainfall using microwave links from commercial cellular communication networks, *Water Resour. Res.*, 47, W12505, <https://doi.org/10.1029/2010WR010350>, 2011.
- Overeem, A., Leijnse, H., and Uijlenhoet, R.: Retrieval algorithm for rainfall mapping from microwave links in a cellular communication network, *Atmos. Meas. Tech.*, 9, 2425–2444, <https://doi.org/10.5194/amt-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 Resour. Res.*, 52, 8039–8065, <https://doi.org/10.1002/2016WR019412>, 2016b.
- Pastorek, J., Fencl, M., Rieckermann, J., and Bareš, V.: Commercial microwave links for urban drainage modelling: The effect of link characteristics and their position on runoff simulations, *J. Environ. Manage.*, 251, 109522, <https://doi.org/10.1016/j.jenvman.2019.109522>, 2019.
- Piczak, K. J.: Environmental sound classification with convolutional neural networks, in: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA, 17–20 September 2015, IEEE, 1–6, IEEE, Boston, MA, USA, 1–6, <https://doi.org/10.1109/MLSP.2015.7324337>, 2015.
- Polz, J.: *cnn_cml_wet-dry_example*, GitHub, available at: https://github.com/jpolz/cnn_cml_wet-dry_example, last access: 2 July 2020.
- Schleiss, M. and Berne, A.: Identification of Dry and Rainy Periods Using Telecommunication Microwave Links, *IEEE Geosci. Remote S.*, 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, *Environ. Res. Lett.*, 12, 034026, <https://doi.org/10.1088/1748-9326/aa5f46>, 2017.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, 15, 1929–1958, available at: <http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf> (last access: 2 July 2020), 2014.
- Uijlenhoet, R., Overeem, A., and Leijnse, H.: Opportunistic remote sensing of rainfall using microwave links from cellular communication networks, *WIREs Water*, 5, e1289, <https://doi.org/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, *Atmos. Meas. Tech.*, 5, 1847–1859, <https://doi.org/10.5194/amt-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, IAHS Publ., 351, p. 7, 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>, 2017.