

Bachelor Project



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Cybernetics**

Rainfall Intensity Estimation from CML Link Data

Petr Novota

**Supervisor: Prof. Dr. Ing. Jan Kybic
May 2022**

Acknowledgements

I would like to thank prof. Dr. Ing. Jan Kybic for his support, expert guidance and numerous valuable ideas. Also, I would like to thank Ing. Martin Fencl, Ph.D. for his last minute tips. Lastly, many thanks to my wife and family who supported me along the way.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with methodical instructions for observing the ethical principles in the preparation of university theses.

Říčany 20. 5. 2022
Petr Novota

Abstract

Since 2006 researches are trying to find a good model for rain intensity estimation based on commercial microwave links **CMLs** signal attenuation data. For a long time models were using a power-law based algorithm. Recent development showed that a recurrent neural network based model can outperform them. Our goal is to investigate different, on neural network based, concept, namely the convolutional neural network **CNN**. We build a wet-dry classification model and analyzed its performance on real data retrieved over 3 years from 28 CMLs located in Prague. Our analysis shows that it can reach F1-scores over 0.98 for rain intensities $I, I > 2 \frac{\text{mm}}{\text{hr}}$ and that it has still room for improvement for $0 \frac{\text{mm}}{\text{hr}} < I \leq 2 \frac{\text{mm}}{\text{hr}}$. Our results are also comparable to the recently published CML model by Polz et al. [16]. In the second part we build a CNN based rain estimation model and its results are competitive with the latest state-of-the-art GRU driven model introduced by Habi et al. [9]. We present different variations of our approach and compare all results. Our research shows that a CNN based model is another promising candidate in the search of an all-round, well performing and CML independent rain estimating model.

Keywords: CML, CNN, GRU, rain intensity estimation, rain binary classification, neural networks

Supervisor: Prof. Dr. Ing. Jan Kybic
Na Zderaze 269/4
místnost: G-104b
Praha

Abstrakt

Od roku 2006 se díky rozvoje telekomunikačních sítí snaží výzkumníci nalézt účinný model pro odhad hustoty deště z útlumu signálu komerčních mikrovlnných spojů. Původ této myšlenky se datuje až do roku 1977. Po dlouhou dobu byly vytvářeny modely založené na mocninné funkci. Nejnovější výzkum ukázal, že modely založené na rekurentní neuronové síti dosahují lepších výsledků. Naším cílem je prozkoumat výkonost modelu založeném na konvoluční neuronové síti **KNS**. Nejprve jsem postavili model klasifikující data do třídy prší a neprší a analyzovali jsme jeho přesnost na reálných datech z 28 CML nacházejících se v Praze. Naše analýza ukazuje že F1-skóre dosahuje hodnoty 0.98 pro intenzity deště $I, I > 2 \frac{\text{mm}}{\text{hr}}$ a že má stále prostor pro zlepšení pro $0 \frac{\text{mm}}{\text{hr}} < I \leq 2 \frac{\text{mm}}{\text{hr}}$. Dále jsme tento model porovnali s obdobným návrhem prezentovaným Polz et al. [16] a dosáhli jsme rovnocenných výsledků. Ve druhé části jsme konvoluční síť upravili tak, aby odhadovala hustotu deště. Její výsledky jsou srovnatelné s nejmodernější neuronovou sítí postavené na architektuře **GRU** představené Habi et al. [9]. Představujeme různé varianty modelu a srovnáváme jejich výsledky. Náš výzkum ukazuje, že KNS modely jsou dalším slibným kandidátem při hledání výkonného, na CML nezávislého modelu pro odhad hustoty srážek.

Klíčová slova: CML, KNS, GRU, odhad hustoty deště, binární klasifikace deště, neuronové sítě

Překlad názvu: Stanovení hustoty deště z data CML spojů

Contents

1 Introduction	3
1.1 Existing methods	4
1.2 Data visualisation	5
1.2.1 General overview	6
2 Methods	13
2.1 Data	13
2.1.1 Commercial microwave links	13
2.1.2 Rain gauges (RG)	13
2.1.3 CML attributes	14
2.1.4 Temperature	14
2.2 Data preparation	14
2.2.1 Attenuation	14
2.2.2 Reference	16
2.3 Evaluation metrics	16
2.3.1 Classification	17
2.3.2 Regression	17
3 Wet-dry classification	19
3.1 Thresholding classification	19
3.1.1 Threshold calculation	20
3.1.2 Results	20
3.1.3 Discussion	20
3.2 Neural network models	21
3.3 Shallow neural network classification model	22
3.4 CNN Base classification model	24
3.4.1 Results	26
3.4.2 Discussion	29
3.5 BCM further experiments	29
3.5.1 Training details	30
3.5.2 Misclassification visualization	32
3.5.3 Results	33
3.5.4 Discussion	34
4 Rain intensity estimation	45
4.1 Architectures	45
4.1.1 The Improved regression model	45
4.1.2 Multi-channel input regression model	46
4.2 Loss functions	48
4.3 Results	49
4.3.1 Discussion	50
5 Software solution	55
Bibliography	59

Figures

1.1 Cml and rain gauges setup	6	3.24 CML 56 wrong predictions mild rain	40
1.2 Heavy rain CML_{56} and all rain gauges data	7	3.25 CML 56 wrong predictions medium rain	40
1.3 Medium rain CML_{56} and all rain gauges data	8	3.26 CML 444 classification training progress SL = 1day, CD = True ..	41
1.4 Mild rain beginning CML_{56} and all rain gauges data	9	3.27 CML 444 classification training progress SL = 1day, CD = False ..	41
1.5 CML 56 general data overview ..	9	3.28 CML 4 ROC with F1-score evolution worst result	42
1.6 CML 105 general data overview	10	3.29 CML 444 ROC with F1-score evolution best result	42
1.7 CML 412 general data overview	10	3.30 CML 444 wrong predictions mild rain	43
1.8 CML 444 general data overview	11	3.31 CML 444 wrong predictions medium rain	43
3.1 Thresholding CML 56 F1-scores	21	4.1 Basic regression model architecture	46
3.2 Thresholding CML 444 F1-scores	22	4.2 Improved regression model architecture inspired by VGG-16 ..	47
3.3 Thresholding CML 547 F1-scores	23	4.3 Multi-channel input model	48
3.4 Conv net classification	24	4.4 Multi-channel scaled MSE loss regression models training progress	50
3.5 Shallow network architecture . . .	24	4.5 Improved regression model using Huber loss	51
3.6 Shallow neural network CML 56 ROC	26	4.6 Improved regression model using Huber loss	52
3.7 Shallow neural network CML 444 ROC	27	4.7 Multi-channel regression model using scaled MSE loss	53
3.8 Shallow neural network CML 547 F1 test scores	28	4.8 Multi-channel regression model using scaled Huber loss	54
3.9 Shallow nn misclassification for low rain intensity	29	5.1 Data class output format	57
3.10 Conv net classification	30		
3.11 Conv. net initial experiment training progress CML 56	31		
3.12 Conv. net initial experiment training progress CML 547	32		
3.13 Conv. net initial experiment training progress CML 444	33		
3.14 CML 547 F1-score distribution	34		
3.15 CML 56 misclassifications	35		
3.16 CML 56 misclassifications	36		
3.17 CML 444 misclassifications	36		
3.18 CML 444 initial conv. net continuous data	37		
3.19 CML 56 initial conv. net continuous data	37		
3.20 CML 56 classification training progress SL = 1day, CD = True ..	38		
3.21 CML 56 classification training progress SL = 1day, CD = False ..	38		
3.22 CML 56 ROC with F1-score evolution worst result	39		

Tables

3.1 CML 56 F1-scores for different classification experiments	25
3.2 F1-score for CML 56 different models comparison	25
3.3 F1-score for CMI 444 different models comparison	26
3.4 F1-score for CML 56 different models comparison	27
3.5 F1-score for CMI 444 different models comparison	28
3.6 CML 444 F1-scores for different classification experiments	32
3.7 Final F1-score for CML 56 different models comparison	34
3.8 Final F1-score for CMI 444 different models comparison	35
4.1 NRMSE regression experiments comparison	49

I. Personal and study details

Student's name: **Novota Petr**

Personal ID number: **373437**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Rain Intensity Estimation from CML Link Data

Bachelor's thesis title in Czech:

Stanovení hustoty dešť z dat CML spoj

Guidelines:

Current standard rainfall monitoring networks don't provide sufficient spatial and temporal resolution or are absent in urban areas. Commercial microwave links (CML) are ubiquitous all around the world. Since rain increases microwave attenuation, it should be possible to detect the rain and estimate its presence in order to build accurate real-time heavy rain warning for urban water management systems. However, the robustness and accuracy of current systems is limited.

Instructions:

1. Get familiar with the available data, curate it and visualize. Prepare also synthetic data for testing.
2. Get familiar with existing methods . Implement baseline subtraction and thresholding and test its performance.
3. Design, implementation and test a shallow neural network on wet-dry classification problem as well as the rain intensity estimation problem based directly on the measurements as well as on preprocessed features.
4. Design, implement and test a convolutional neural network for the same problems.
5. Design, implement and test a recurrent neural network on for the same problems.
6. Extend the best performing methods to use multiple CMLs and to estimate the rain density at multiple points. The spatial extension can be designed analytically or learned.

Bibliography / sources:

- [1] M. Fencel, P. Valtr, M. Kvi era and V. Bareš, "Quantifying Wet Antenna Attenuation in 38-GHz Commercial Microwave Links of Cellular Backhaul," in IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 4, pp. 514-518, April 2019, doi: 10.1109/LGRS.2018.2876696.
- [2] Fencel, M., Dohnal, M., Rieckermann, J., and Bareš, V.: Gauge-adjusted rainfall estimates from commercial microwave links, Hydrol. Earth Syst. Sci., 21, 617–634, <https://doi.org/10.5194/hess-21-617-2017>, 2017.
- [3] H. V. Habi and H. Messer, "Recurrent Neural Network for Rain Estimation Using Commercial Microwave Links," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 5, pp. 3672-3681, May 2021, doi: 10.1109/TGRS.2020.3010305.
- [4] 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.

Name and workplace of bachelor's thesis supervisor:

prof. Dr. Ing. Jan Kybic Biomedical imaging algorithms FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **10.02.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

prof. Dr. Ing. Jan Kybic
Supervisor's signature

prof. Ing. Tomáš Svoboda, Ph.D.
Head of department's signature


prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature



Chapter 1

Introduction

The goal of this work is to investigate different approaches on how to estimate rain intensity based on commercial microwave links (CML) signal attenuation (A).

CMLs are point-to-point radio connections widely used as cellular backhaul. A substantial part of CML networks is operated at frequencies between 20 and 40 GHz where radio wave attenuation caused by raindrops is almost proportional to rain intensity [15]. These CMLs can, therefore, be used as unintended rainfall sensors providing path-integrated quantitative precipitation estimates. Moreover, CML data are accessible online in real time from network operation centers either through network monitoring systems or specifically designed server-sided applications [5].

The idea of deriving precipitation estimates from microwave signal attenuation was originally proposed over 40 years ago [2]. Recently, it grew on popularity due to an extensive growth of cellular networks [12], [10] which frequently incorporate CMLs. Currently, there are about four million CMLs being used worldwide and their number is increasing.

The reason why it is interesting to use CMLs for rain precipitation intensity estimation is that rainfall data of sufficient quality is lacking for most of the Earth's land surface. Moreover, coverage by classical surface precipitation gauging networks is declining in many regions around the world [11]. Although global precipitation data can be obtained from satellites or local monitoring networks, their spatiotemporal resolution is still insufficient for hydrological modeling of small, mountainous or urban areas [11].

The high requirement for good temporal and spatial resolution of rainfall data is especially prominent in urban catchments, because they differ from natural ones in two major aspects. The examined area scale in urban and natural catchments differs in orders of magnitude and the former is covered by a high ratio of impermeable surfaces that limit rainfall infiltration and lead to more surface runoff.

Tipping bucket rain gauges stand for traditional way of retrieving precipitation measurements in urban areas. However, they often fail to supply sufficient spacial and temporal resolution, often due to the low rain gauge counts. In particular, when heavy storm events, crucial for the evaluation of urban stormwater systems, are considered, the spatial representativeness of

point rainfall observations from rain gauges is limited.

1.1 Existing methods

The beginning of rain intensity estimation from CMLS dates in 2006 when Messer et al. [12] introduced the use of a single CML as a mean to estimate rain intensity. Later, Leijnse et al. (2008) [10] used, for the first time, two CMLs in the Netherlands. Today, researches work with up to 3900 CMLs.

One of the challenges at the beginning was the acquisition of CML data. In 2012, Chwala et al. [6] first acquired data using loggers at CML towers. He later introduced a state-of-the-art CML data acquisition for research [5] and improved hereby the data availability for other researches.

The Wet antenna effect (**WAA**) 2.2.1 was initially estimated as a constant. Later studies showed that its magnitude ranges from 1.5dB - 9dB [14]. Therefore considering it as constant led to significant overestimations of peak rainfalls [8]. Moreover, the WAA magnitude can either be quantified from short CMLs, where it dominates the rain induced attenuation, but it can be also retrieved from long-term statistics of CMLs and rainfall climatological data [8].

The main methods at the beginning were relying on the power law model (**PL**) 1.1, where a and b are the PL coefficients. In 2020 Habi et al. [9] used a recurrent neural network and showed that this method beats the PL performance. Its estimation error is lower by up to 50%. The estimation error is estimated for different rain intensities and it ranges from $1.5I_{avg}^i$ for $0 \frac{\text{mm}}{\text{hr}} < I \leq 1 \frac{\text{mm}}{\text{hr}}$ down to $0.5I_{avg}^i$ for $15 \frac{\text{mm}}{\text{hr}} < I \leq 25 \frac{\text{mm}}{\text{hr}}$. I_{avg}^i represents the average rain intensity in the given interval.

$$A_r(t) = aR^b(t)l \quad (1.1)$$

There has been a lot of work done on this topic and the latest paper showed that a neural network based model can deliver better performance for rain rate estimation than previously used models. At the beginning, researches developed complicated mathematical models and tried to find the correct formulas describing all attenuation effects. Today, the focus has shifted towards neural networks. They can take the burden of finding those formulas off our shoulders. We just feed them with enough data and they figure it out on their own. But of course, the reality with them is not that simple and there are many problems to be solved until they work really well. And thus we would like to expand our understanding of neural network models used for rain intensity estimation.

We will proceed in two steps. In the first step we will build Wet-dry classification model. Its goal is to classify given A or a A sequence into Wet class or Dry class. Wet class means that it is raining and Dry class means it is not raining. We will compare our results to the results published in recent paper published by Polz et al. in 2020 which used a convolutional

neural network model [16] which is very similar to ours. We will repeat their experiments with different data set and confirm their results.

In the second step we will expand our models so that they are capable of rain intensity (**I**) estimation. We will compare them between each other and with the state-of-the-art model published by Hai Victor Habi and Hagit Messer in May 2021 [9]. As model backbone, they have used two gated recurrent units (**GRU**). We will use a CNN based architecture. We hope either to find a new perspective approach that can be further developed or to exclude it from further research.

1.2 Data visualisation

First of all we want to get an overall picture about the data we will work with. The CML and RG spatial setup is depicted in figure 1.1. Red lines represent CML links and rainy clouds stand for RGs. CML link 56 is yellow. Let us illustrate how correlation between CML_{56} and RGs looks like. The highest correlation is expected with D_{10} . We classify the rain intensity (**I**) in four classes. Heavy (**H**), medium (**M**) and mild (**L**)rain and no rain(**O**) like this 1.2

$$\begin{aligned}
 I(t) = D_{avg}(t) &= \frac{D^{10}(t) + D^{13}(t) + D^{22}(t)}{3} \\
 H &= \left\{ I(t) \mid I(t) > 40 \frac{\text{mm}}{\text{hour}} \right\} \\
 M &= \left\{ I(t) \mid 10 \frac{\text{mm}}{\text{hour}} < I(t) < 15 \frac{\text{mm}}{\text{hour}} \right\} \\
 L &= \left\{ I(t) \mid 1 \frac{\text{mm}}{\text{hour}} < I(t) < 1.2 \frac{\text{mm}}{\text{hour}} \right\} \\
 O &= \left\{ I(t) \mid I(t) < 1 \frac{\text{mm}}{\text{hour}} \right\}
 \end{aligned} \tag{1.2}$$

Heavy rain situation in figure 1.2 shows that correlation with rain gauge D_{10} is very high. At the beginning of the rain the CML_{56} seems to react faster than D_{10} . Other rain gauges detect the rain as well, but there is a time shift with regards to the D_{10} and their data has different shape. Overall this data shows, that dry wet classification for heavy rain events should be doable. But it is a rare situation, there are total of 1 092 179 data points and only 85 can be classified as H according to 1.2

Observation of a medium rain episode figure 1.3 shows that the correlation between A and RGs is not that high in many situations. D_{10} starts to measure rain ca. 20 minutes after the attenuation of CML_{56} starts to increase. One explanation is that a rain gauge measures precipitation for one small location. The CML signal attenuation, on the other hand, is influenced by the state of weather along a line, which for CML_{56} is 3.195m long.

Another phenomenon can be seen in figure 1.3 and that is the wet antenna attenuation (WAA). So when it stops raining, there will be higher attenuation

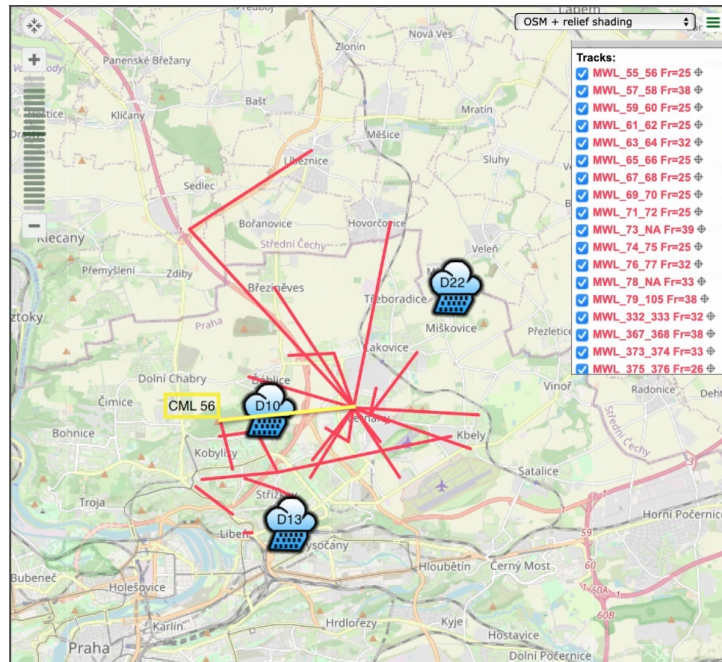


Figure 1.1: Cml and rain gauges setup

until the antenna dries out. This process can take different amount of time as it is influenced by a variety of factors, temperature, humidity, wind, just to name a few.

The next problem is shown in figure 1.4. Due to the point like nature of the RGs action radius it is usual for A to increase sooner then RG detects rain. In this particular case there is a time shift of approx. 20 minutes.

From the data we have just shown can be concluded that the correlation between A and RGs is positive, but as discussed, it is sometimes higher and other times lower depending on the rain event nature.

1.2.1 General overview

In the next part we want to get an overall picture of A development for multiple CMLs. Attenuation data for CML_{56} in figure 1.5 show a long time period around June 2014 where the A_b^{56} suddenly increases from 58db to 72db and then decreases very slowly over the period of several months until it reaches the original value in October 2014. In general, the A_b can change due to an obstacle placed on its path, that would explain a sudden increase, but we have no explanation for the a decrease.

CML_{105} data in figure 1.6 from the same time period show a similar event where in May 2014 $A_b^{105} = 53\text{dB}$ and goes up to $A_b^{105} = 56\text{dB}$. But this could be explained by a new obstacle being placed on the CML path because A_b^{105} does not return to its original value as it is the case for CML 56. Other CMLs like 412 in figure 1.7 don't show such a behavior.

Another interesting A development happens for CML_{444} in figure 1.8. The

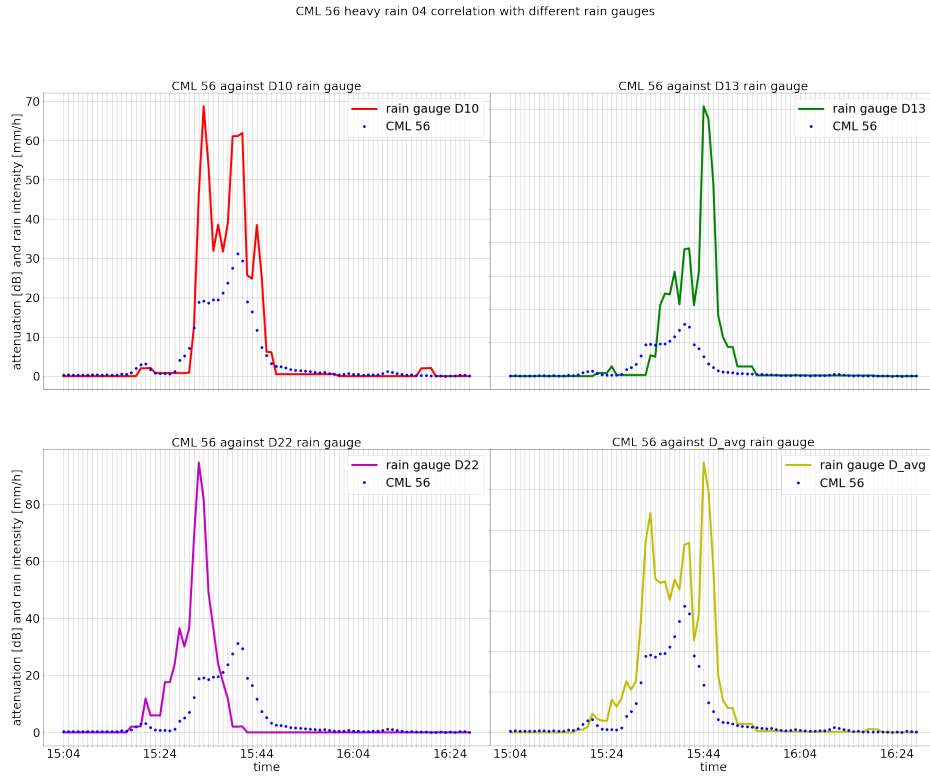


Figure 1.2: Heavy rain CML_{56} and all rain gauges data

A_b^{44} makes 20dB up and down jumps. This could still be explainable by obstacles being placed and removed from the CML path.

The CML 56 behavior, where the attenuation baseline suddenly jumps up and then slowly decreases is not seen in other data.

We showed that the attenuation data is not always a steady baseline with increased A during rain periods. We see that the baseline can change either gradually or step wise and might or might not come back to its original value.

CML 56 medium rain 01 correlation with different rain gauges

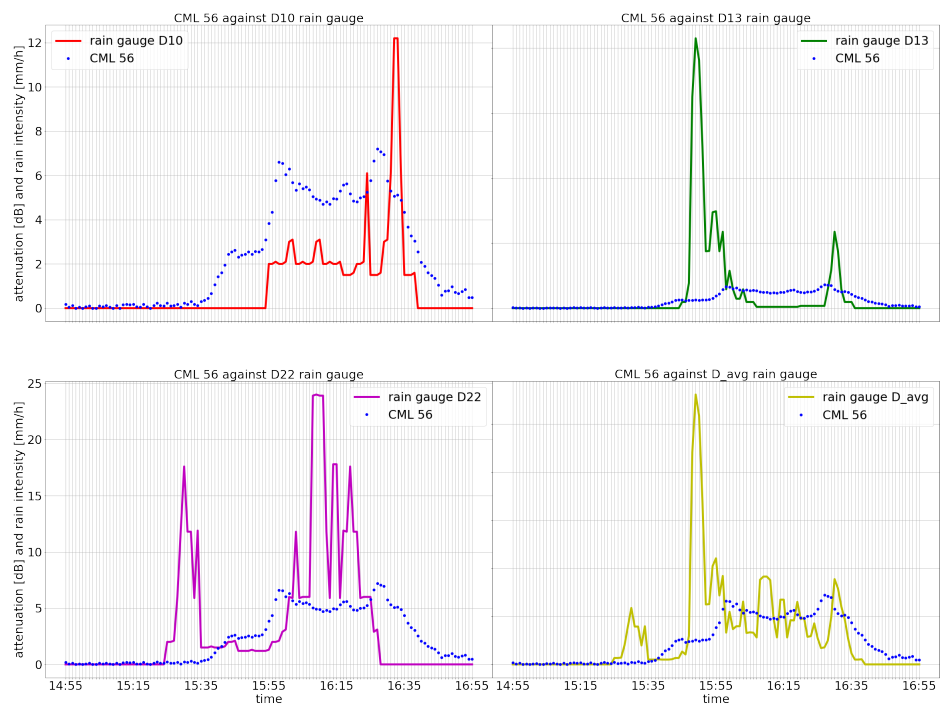


Figure 1.3: Medium rain CML_{56} and all rain gauges data

CML 56 mild rain 02 correlation with different rain gauges

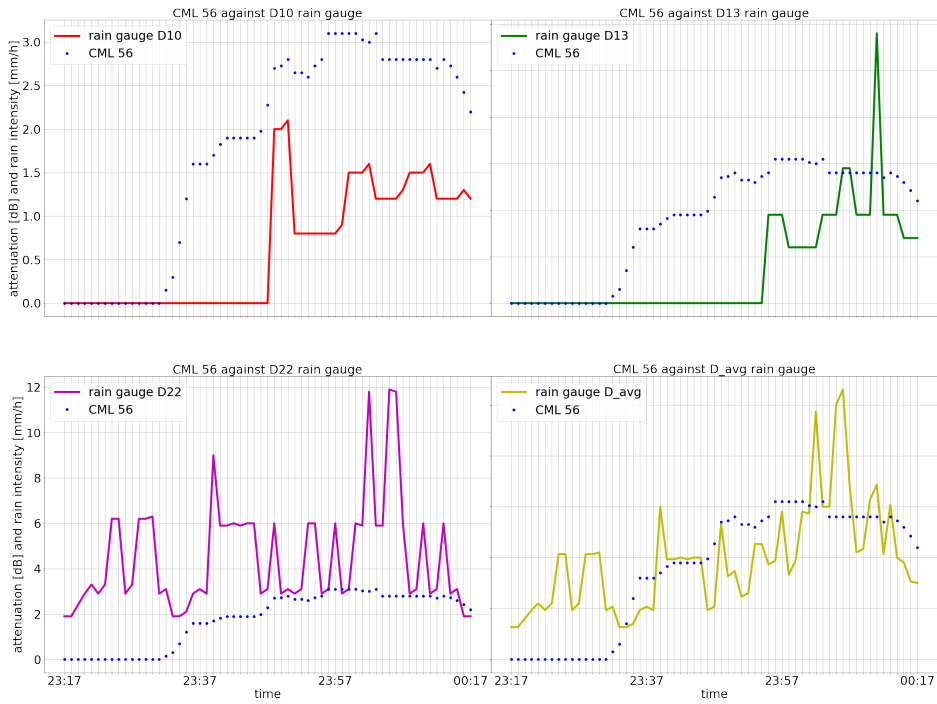


Figure 1.4: Mild rain beginning CML_{56} and all rain gauges data

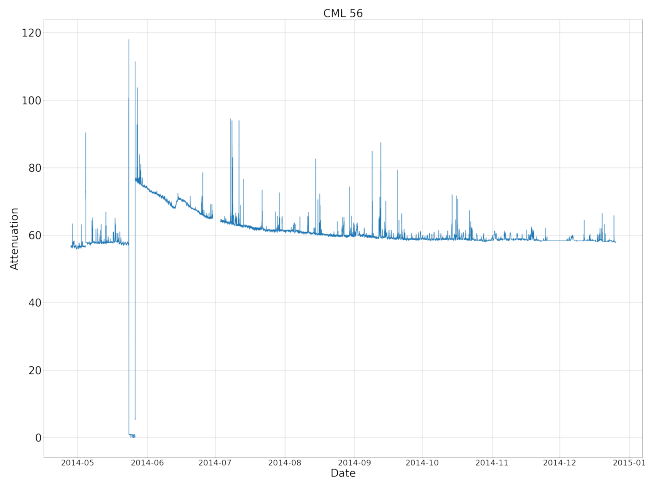


Figure 1.5: CML 56 general data overview

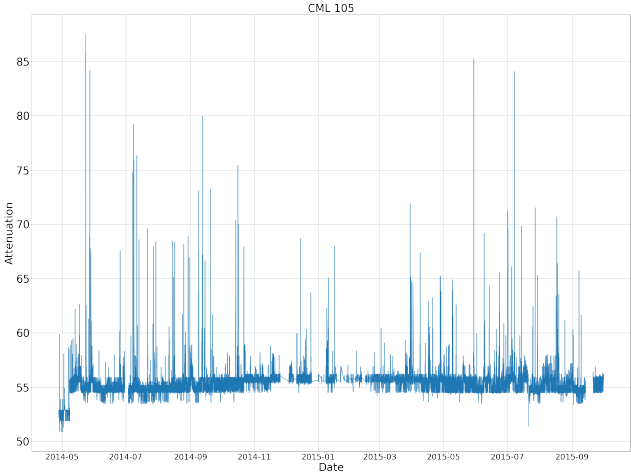


Figure 1.6: CML 105 general data overview

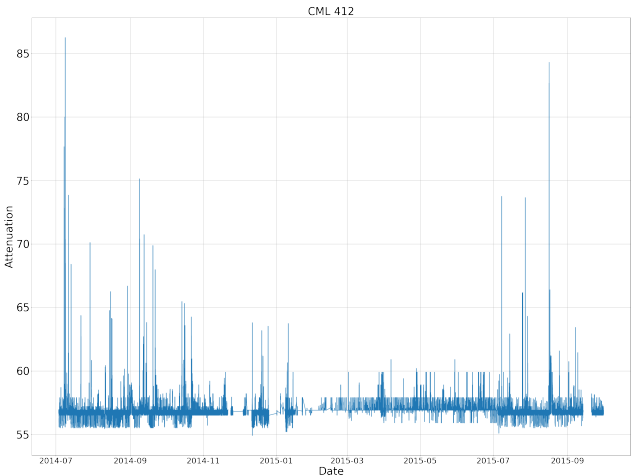


Figure 1.7: CML 412 general data overview

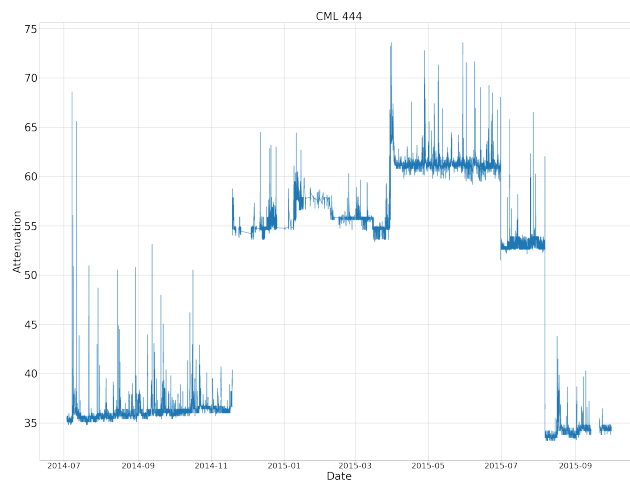


Figure 1.8: CML 444 general data overview

Chapter 2

Methods

Lorem ipsum

2.1 Data

All data and scripts used in this work are located on Unix computers of the Department of Cybernetics in the /datagrid/Medical/tel4rain folder.

2.1.1 Commercial microwave links

This work is based on data from 28 CMLs working in full duplex mode. For each CML link we have a sequence of triplets. There are transmitted signal power (P_{tx}), received signal power (P_{rx}) and timestamp (\mathbf{t}). The data span from the end of April 2014 to half of January 2017. There is one csv file for each CML link in form of a time series with various sampling frequencies ranging from 0.5s to 15s. The P_{rx} is in dB and it is stored as a decimal number with one significant digit and quantization of 1/3dB. The P_{tx} is also in dB and it is stored as an integer. Its quantization step is 1 dB.

Most of the links have automatic transmit power control (**ATPC**). This feature increases P_{tx} during attenuation events in order to keep P_{rx} constant (or until max. P_{tx} is reached). The configuration of ATPC, however, differs significantly from device to device. Some keep the P_{tx} constant, some changes step wise when P_{rx} threshold is reached (thus, sometimes only during stronger attenuation events), and some keep P_{tx} constant, i.e. behave as being without ATPC. As P_{rx} and P_{tx} quantizations differ (1/3 dB, resp. 1 dB), the specifics of ATPC configuration (P_{tx} constant vs. P_{tx} compensating P_{rx}) affect quantization of obtained signal power data. We will not have to worry about it, because we will work only with signal attenuation (**A**) at time \mathbf{t} defined as:

$$A(t) = P_{tx}(t) - P_{rx}(t) \quad (2.1)$$

2.1.2 Rain gauges (RG)

We have three RGs, D_{10} , D_{13} and D_{22} . They measure rain intensity in

millimeters per hour over an interval of 1 minute. That means their time resolution is 1 minute. We also know each RG's position so that it's relevance for a given CML can be evaluated.

RG's have sometimes breakdowns or report non-valid data, therefore such data must be discarded.

■ 2.1.3 CML attributes

Each CML has multiple attributes. Most useful ones for model training are CML length (\mathbf{l}), frequency (\mathbf{f}), polarization ($\boldsymbol{\alpha}$) and position (\vec{p}). Because we aim to create on CML independent model, CML attributes can be used as model inputs and thus providing information about the particular CML. The most important attribute is the CML length, because the rain induced attenuation is proportional to it [4]. Other attributes might have some effects too and a neural network model could detect them and use them for learning. But we don't have enough different attribute combinations so we will only use the CML lengths.

■ 2.1.4 Temperature

From CHMU Prague we have got temperature measurements (τ) for Prague - Prosek. Their time resolution is one minute. We use this data to ensure that we only estimate liquid precipitation. In order to ensure this, we only use data for which $\tau > 5^\circ$. In order to estimate snow or sleet intensities, different models would have to be build, because they affect A differently. Once a good liquid precipitation estimating model is known we can, with enough data, build models for the remaining precipitation types.

■ 2.2 Data preparation

TODO:

■ 2.2.1 Attenuation

CMLs provide data samples with sampling frequency ranging from 0.5s^{-1} up to 4min^{-1} . There are time periods where data samples are missing or where P_{rx} , P_{tx} reported erroneous values. Therefore, non-valid data are removed and attenuation is calculated 2.1.

$$A = A_b + A_{wa} + A_r + \epsilon \quad (2.2)$$

CML Attenuation (\mathbf{A}) is composed of four parts 2.2. The Base attenuation (\mathbf{A}_b) represents the signal scattering on air molecules along the CML path. It is the attenuation that is always there. The wet antenna attenuation (\mathbf{A}_{wa}) [10], [17] is caused by the antenna being covered by water. The next part is the rain induced attenuation (\mathbf{A}_r), which is proportional to l . Lastly, ϵ stands for unknown attenuation effects.

■ Attenuation time resolution

The next thing to consider is what attenuation time resolution A_{res} do we want to use. Multiple previous works [9], [7] had $A_{res} = 15\text{min}$. This means that two data samples are 15 minutes apart. We think that this is not fast enough. If a heavy rain starts, then we want to be notified as soon as possible. For this reason we will use $A_{res} = 1\text{min}$. This results in total of n data samples which define a set $N^c = \{x \in \mathbb{N} \mid x \leq n\}$ for each CML \mathbf{c}

The next aspect to choose is how do we aggregate A samples in order to reach the chosen A_{res} . In this work, the mean value will be used. In this recent article [9] the maximum function was used. But this can go wrong if there is one outlier in the sequence. Therefore we choose the mean.

As result there is a set of points S^c for each CML that can be used as input for different models.

$$S^c = \{(t, A^c(t)) \mid t \in N^c\} \quad (2.3)$$

■ Continuous data

CML and RG measurements have sometimes missing data resulting in time gaps. Sometimes there is non-valid data reported, sometimes there is no data available for certain time periods, because the measurement device was not working. For model training, which takes last n samples as input, it makes sense to only use n time consecutive samples. If a model uses $n = 1440$, we should make sure that there is no gap and thus no part of the input data is too old and not relevant. We considered two ways how to deal with it.

On one hand, we can ignore this problem. Because there is no rain around 97% of time, there is a big chance that such a gap occurs in a no-rain period. As shown in Data visualisation the A_b does not usually change across several days. This means that a data gap during a no-rain period will likely not negatively influence a model's training.

On the other hand, we can make sure that data is always continuous. For all S^c we define a maximal time window W_{max} and a time gap \mathbf{G} . Then we iterate over all points in S^c and create subsets S_i^c as follows 2.4

$$\begin{aligned}
 (1) & \quad j = 0; \quad n = |N^c| - 1 \\
 (2) & \quad S_j^c = S^c \\
 (3) & \quad a = S_0^c[0] \\
 (4) & \quad \text{while } i < |N^c| \\
 (5) & \quad \text{if } t_i - t_{i-1} \leq W_{max} \Rightarrow A_{new} = \frac{A_i - A_{i-1}}{2} \\
 (6) & \quad \text{else } S_j^c = \{t_a, \dots, t_{i-1}\} \wedge S_{j+1}^c = \{t_i, \dots, t_n\} \\
 (7) & \quad j ++; \quad a = S_j^c[0]; \quad i ++
 \end{aligned} \quad (2.4)$$

In case the gap is small enough, A_{res} number of A_{new} values is created and the gap is filled with them. We chose $W_{max} = 3$

In our experiments we will show model performance for both data preparation methods.

2.2.2 Reference

The next task is to prepare references for the training. Two references will be needed. One for classification and second one for regression.

Classification reference

Reference is prepared for classification tasks as follows. From each RG (D^d) $d \in \{10; 13; 22\}$ there is one classification reference (C_{ref}^d) generated where d denotes the RG number. We define thresholds $T_{low} = 0.1$ and $T_{high} = 0.5$ and time interval $i = 15\text{min}$. Reference is calculated as follows 2.5.

$$M^d(t) = \frac{1}{i} \sum_{t-i}^t D^d(x)$$

$$C_{ref}^d(t) = \begin{cases} 0, & \text{if } M^d(t) < T_{low} \\ 1, & \text{if } M^d(t) > T_{high} \\ -1, & \text{else} \end{cases} \quad (2.5)$$

The gap between T_{low} and T_{high} means that we are not sure whether it is raining or not and by labeling it with -1 we can exclude it during model training and evaluation.

Regression reference

Similarly to classification reference, from each RG one regression references (R_{ref}^d) is created. The R_{ref} corresponds to RG measurements as follows 2.6

$$R_{ref}^d(t) = D^d(t) \quad (2.6)$$

2.3 Evaluation metrics

We will present results for CMLs 56, 444 and 547. This allows us to compare all experiments. Experiments with neural network models will take long time and it would not be time efficient to include all CMLs. For this reason we chose three CMLs. This allows us to try a lot more experiments, compare them between each other and when a promising model is found, then we can use it on more CMLs. CML 56 was chosen, because it is very close (122m) to the RG D_{10} . It's length is 3,196km, CML 444 is near RG D_{13} (926m) and it's length is 0,854km and CML 547 is far from the nearest RG D_{10} (1.500m) with length 1,086km.

In order to evaluate our models we will use the F1-score for classification models and root mean square error (**RMSE**) and normalized RMSE (**NRMSE**) for regression models. The RMSE is chosen, because it is used to evaluate the GRU network [9] and we want to compare our results with it.

■ 2.3.1 Classification

■ F1-score

There are several ways how to evaluate classification model performance. In our case classes are heavily unbalanced. It is not raining 97.2% of time, so a model which would always predict "no rain" would have very high accuracy. That would look great, but it would not be. Therefore, mainly the F1-score (F_1) 2.9 will be used to evaluate all classification models.

In order to compute F_1 , precision (**PPV**) and recall (**TPR**) have to be calculated. Precision accounts for true positive (**TP**) and false positive (**FP**) predictions. Recall accounts for true positive and false negative (**FN**) predictions. By true positives we understand a rain episode correctly classified. In this way we can get a sense of the model overall performance. If model predicts no rain everywhere, it will have ca. 97% accuracy but f1 score would be zero.

$$PPV = \frac{TP}{TP + FP} \quad (2.7)$$

$$TPR = \frac{TP}{TP + FN} \quad (2.8)$$

$$F_1 = \frac{2 * PPV * TPR}{PPV + TPR} \quad (2.9)$$

■ Matthews correlation coefficient

The second metric we will use for classification model evaluation is the Matthews correlation coefficient **MCC**. It is a commonly used metric for binary classification [3]. It is acknowledging the possibly skewed ratio of the wet dry periods and it is high only when the classifier performs well on both of those classes. $MCC = 0$ represents a random guessing and $MCC = 1$ represents a perfect classifier. A strong correlation is given at values above 0.25 [1]. This metric was also used by Polz et al. so it will allow for results comparison.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \in [-1, 1] \quad (2.10)$$

■ 2.3.2 Regression

■ RMSE

We want to compare our model with the recurrent network GRU used in Habi et al. [9], therefore we will use the same regression evaluation. They

were using two metrics. One of them is the Root mean square error (**RMSE**) calculated as follows: 2.11

$$RMSE(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{1}{N} \sum_i (x_i - \hat{x}_i)^2} \quad (2.11)$$

where \mathbf{x} is the reference vector, $\hat{\mathbf{x}}$ is the estimates vector and N is the number of samples.

■ **NRMSE**

The second metric is the normalised RMSE (**NRMSE**) defined as: 2.12

$$NRMSE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{RMSE(\mathbf{x}, \hat{\mathbf{x}})}{\frac{1}{N} \sum_i x_i} \quad (2.12)$$

Chapter 3

Wet-dry classification

In this part the goal is to create an accurate wet-dry classification model, that is a model that predicts whether it is raining at given moment or not. As learning reference the C_{ref}^d with time window of 15 minutes is chosen 2.5. The reason we don't pick 1 minute time window, which would correspond to the CML data time resolution, is that we want to compensate for the time shifts between A and C_{ref} . For scenarios where rain intensity is low and changing often, using a longer time window can simplify the reference. Then, it is less influenced by recent fluctuations and reflects an overall trend better.

3.1 Thresholding classification

Looking at eq: 2.2, in this approach we aim to determine the (A_b^c) , c denotes the CML number, in a one step method [13]. We do it by taking the minimal value from a set of consecutive samples as the baseline component 3.1.

$$A_b^c(t) = \min(A^c(t), A^c(t-1), \dots, A^c(t-N_{DB})) \quad (3.1)$$

where N_{DB} is a hyperparameter that sets the number of consecutive samples in the set with sampling frequency . $N_{DB} = 2 \times 64 \times 60$, that is two days.

Once A_b^c is found, the rest of attenuation consists of A_r and A_{wet}^c . Because this first approach is very simple, the A_{wet}^c is neglected. Another reason why neglecting WAA might not be a problem is the fact that it affects attenuation only when it is raining, after it stopped and during the antenna radome drying out. But the fact that classification reference is based on the maximal value from last 15 minutes results in rain reference stating that it rains up to 15 minutes after it has stopped raining. It follows that when not determining rain intensity, where WAA has its effect, but only doing classification should allow us to neglect it. Therefore A_r^c is calculated as

$$A_r^c(t) = A^c(t) - A_b^c(t) \quad (3.2)$$

And classification calculated like this: 3.3

$$C^c(t) = \begin{cases} 0, & \text{if } A_r^c(t) < T^c \\ 1, & \text{else} \end{cases} \quad (3.3)$$

where T is the threshold parameter that needs to be found for each CML and c denotes the CML number. Then finding the threshold is an optimization problem where such T^c is searched that solves eq. 3.4

$$\begin{aligned} \min(g(C^c(t), C_{ref}^c(t))) \\ g^c(C^c(t), C_{ref}^c(t)) = \sum_{t=0}^n |C^c(t) - C_{ref}^c(t)| \end{aligned} \quad (3.4)$$

3.1.1 Threshold calculation

At this point, classification references are prepared and attenuation data have baseline attenuation subtracted. The data is split into 70% train and 30% test data. The best possible T^c is found in two steps.

In the first step, a coarse search on train data is performed. We create fifty evenly spaced values between -10 and 10 and each one of them is used as T^c and F1 score is calculated. The one resulting in best F1-score is saved for the second step

$$step_size = \frac{20}{50} = 0.4 \quad (3.5)$$

In the second step, a fine search is done. There are one hundred evenly spaced values between $best_step_one_value - step_size$ and $best_step_one_value + step_size$. Each of them is used as T^j and F1-score is calculated. The best one is saved and it is the best T^j found. For the best threshold the F1-score on test data is calculated and saved.

This is done for all CMLs in combination with all references, C_{ref}^{10} , C_{ref}^{13} , C_{ref}^{22} . For each reference the best F1 score is saved and at the end, the reference with the best F1 score is saved. All this is done in *thresholding.py* script. It will be interesting to see, which reference is the best and whether the best reference for CML 56 the D_{10} is.

3.1.2 Results

Figures 3.1, 3.2 and 3.3 show thresholding results. Plots show F1-score for data where the $R_{ref} \geq MRI$. MRI is the Minimal rain intensity. In case of CML 56 and 547 results show, that the most failed predictions arise for $0 \frac{\text{mm}}{\text{hr}} \leq R_{ref} \leq 1 \frac{\text{mm}}{\text{hr}}$. Another observation is that for $R_{ref} \geq 2 \frac{\text{mm}}{\text{hr}}$ there is a F1-score downwards trend. Results for CML 444 are bad in comparison. For this particular CML the thresholding did not work at all.

3.1.3 Discussion

It is interesting that results for CML 444 are so bad in comparison with other two CMLs. The explanation might be what we saw in figure 1.8. There are

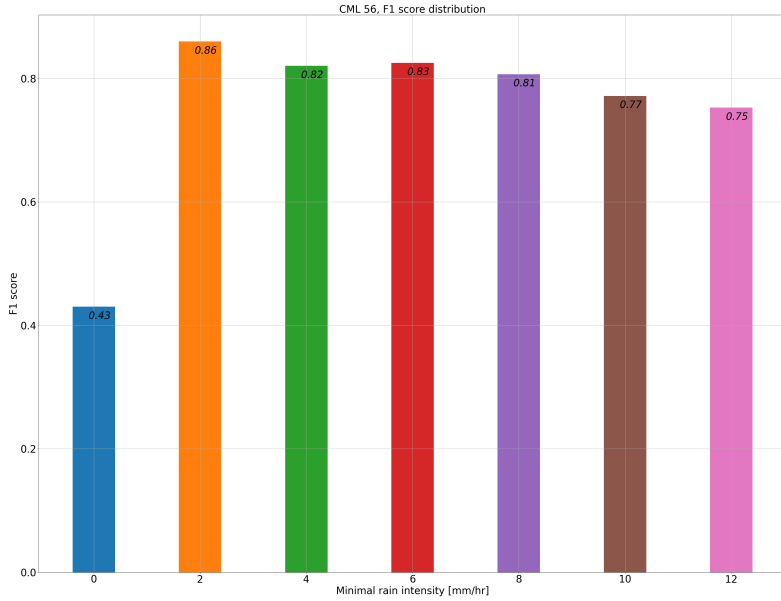


Figure 3.1: Thresholding CML 56 F1-scores

big changes in A_b^{444} . Therefore our simple A_b estimation method might have failed in capturing such a wild dynamic.

Because this is our first experiment we don't have anything to compare it with. The only conclusion is that CML 444 has a bad result and the other two have similar results. Also, the thresholding does not work very well for low precipitation intensities.

3.2 Neural network models

We think that a successful model must include past values in order to predict current precipitation intensity well. Our inspiration for further experiments comes from Habi et al. paper [9]. Their network architecture consists of a Backbone, in the Backbone there are two gated recurrent units (**GRU**) and a shallow linear network for CML attribute processing. Those two parts are concatenated and another shallow linear networks perform classification and regression 3.4.

They use attenuation sampling frequency of 10 seconds, each sample is the maximum over 90 consecutive measurements, i.e. 15minutes. Although the sampling frequency is quite high, we think that taking the maximum from past 15 minutes is a too large time window. Our sampling frequency is 1 minute but each sample is the average of 3 to 90 consecutive measurements, depending on CML, taken from last 1 minute.

We want to change the Backbone network. Our idea is to use a 1D

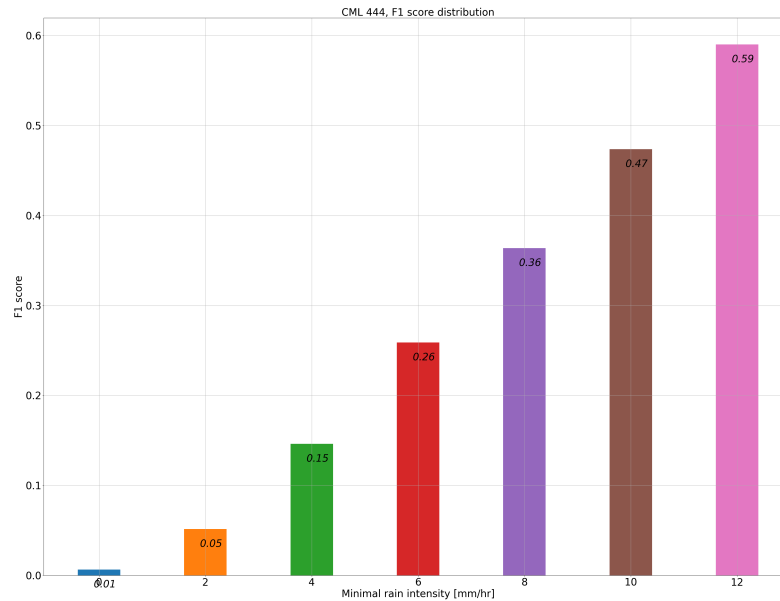


Figure 3.2: Thresholding CML 444 F1-scores

Convolution on n consecutive samples, where optimal n will be determined by experiments. CNNs have been successful in extracting features from pictures. In the rain estimation problem we also need to extract features from past data. In order to find A_{wa} , we need to know, whether it is raining or not and what the intensity range is. That could be two features. Next one could be the A_b . Another feature could describe the rain intensity trend. And there might be a lot of our features that a neural network could extract and we are not even aware of them. This seems similar to feature extraction from pictures, therefore, we will test if a CNN can have equal or better performance for a rain estimation problem then a recurrent network.

We will start with simple networks and gradually build more complex ones.

3.3 Shallow neural network classification model

In the first experiment we will use a shallow neural network for rain classification. We will train one model on single CML.

Network architecture

The network consists of 3 fully connected layers. At time t the input is: $\text{Input} = \{S, \min(S), \max(S), \text{mean}(S), \text{std}(S)\}$ where $S = \{A_t, A_{t-1}, \dots, A_{t-n}\}$

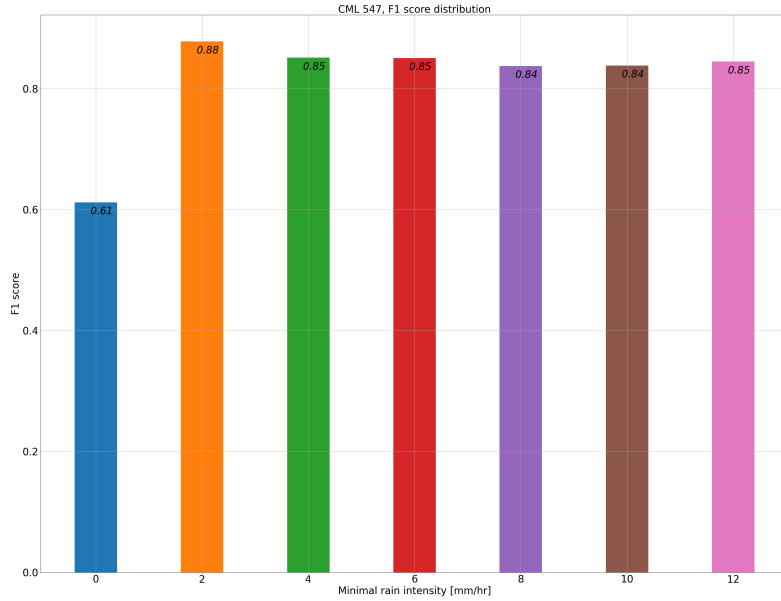


Figure 3.3: Thresholding CML 547 F1-scores

and n is the sample length of two days $n = 24 \times 60 \times 2 = 2880$. The first layer outputs $n + 4$ values, the second one 256 and the last one 2. Between every layer, there is a LeakyRelu activation function. Network architecture is depicted in figure 3.5. Cross entropy loss is used as a loss function.

Results

Results show again that predictions for low I are the least accurate. The F1-score distribution for CML 56 3.2 shows F1-score = 0.62 for $I > 0 \frac{\text{mm}}{\text{hr}}$. We observe a F1-score increase for all three CMLs, two of them are tracked in tables. CMLs 56 and 547 have similar results, therefore we track only one of them. CML 444 shows worst results so we will track its performance separately.

Next results we can see are the ROC curves for CML 56 in figure 3.6 and 444 in figure 3.7. We define Classification threshold (C_{thres}) which is a measure of certainty δ of the models prediction for rain event must be so that the model classifies it as rain. If $\delta > C_{thres}$ then the resulting predictions is rain. We can see that for $C_{thres}^{56} = 0.5$ the False positive rate (FPR) $FPR^{56} = 0.20$ and $C_{thres}^{444} = 0.5$ for $FPR^{444} = 0.50$. Another fact is that for $C_{thres}^{444} > 0.3$ the F1-score goes down.

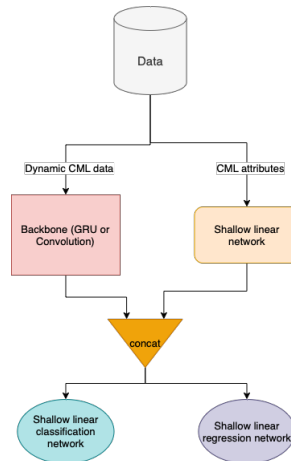


Figure 3.4: Conv net classification

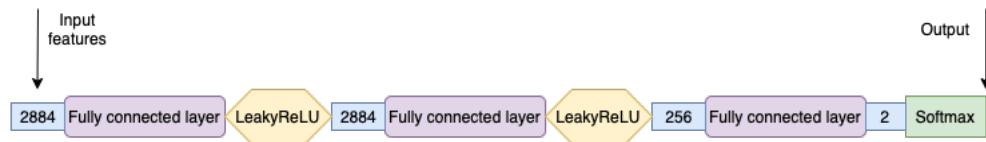


Figure 3.5: Shallow network architecture

Discussion

Data show that a Shallow neural network has better performance than the Thresholding method. It shows better performance for all three CMLs for all I_s . It was also able to get reasonable results for CML 444. It suggests that even a CML with large A_b changes could be usable for model training. It could even help models to generalize better. The biggest problem so far is the performance for low I . Next figure 3.9 depicts a misclassified situation where the rain I is low. All curve offsets are adjusted so that they don't overlap. We can see that the prediction corresponds well with the CML attenuation data but the correlation between them and C_{ref} is not high.

3.4 CNN Base classification model

We build a CNN based classification model called the Base classification model **BCM**. The main difference between BCM and the Polz et al. model is the input sample size. We used one and two days and Polz et al. used 60min to 300min. Another noticeable difference is that we introduced skip-connections after each convolutional layer. And a small difference is also in the linear network where we use three fully connected layers instead of two.

CNNs gained popularity around year 2014 due to their good results in image classification. Today, they are the goto solution for most image processing

			minimal rain intensity [mm/hr]						
SL	CD	A_{res}	0	2	4	6	8	10	12
1 day	False	1 min	0.66	0.97	0.97	0.97	0.97	0.96	0.97
1 day	True	1 min	0.65	0.99	0.98	0.98	0.98	0.98	0.97
2 days	False	1 min	0.72	0.97	0.97	0.98	0.98	0.98	0.98
2 days	True	1 min	0.65	0.99	0.97	0.97	0.97	0.96	0.97
1 day	False	2 min	0.68	0.98	0.98	0.98	0.97	0.97	0.96

Table 3.1: CML 56 F1-scores for different classification experiments

CML 56 F1-score for different minimal rain intensities							
Model \ $I[\frac{mm}{hr}]$	≥ 0	≥ 2	≥ 4	≥ 6	≥ 8	≥ 10	≥ 12
Thresholding	0.43	0.86	0.82	0.83	0.81	0.77	0.75
Shallow nn	0.62	0.95	0.94	0.94	0.94	0.94	0.95

Table 3.2: F1-score for CML 56 different models comparison

problems and show good results for time series as well. The intuition behind their workings is that they are able to extract meaningful features from data like horizontal edges, vertical edges and so on, and combine them into more complex features.

The rain classification problem is in a way similar to this. What we need is to extract attenuation baseline and WAA values and the information whether it is currently raining or not. Polz et al. in 2020 used a CNN based model for wet-dry classification [16]. We will use a slightly modified version, train it on our dataset and compare results. Our experiments show comparable results. Moreover, they exhibit a better FPR to FNR ratio.

■ Architecture

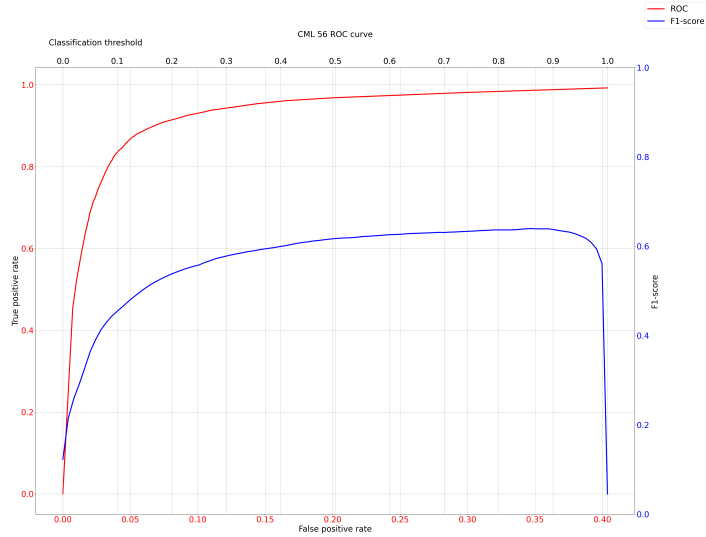
Our model consists of two parts. A CNN and a linear neural network build from three fully connected layers. The convolutional part extracts features from past data and fully connected layers perform the wet-dry classification. The architecture diagram shows how the network looks like figure 3.10.

■ Convolutional layers

The convolutional part of the network consists of eight 1D convolutions. Each one uses kernel with $size = 5$, $stride = 1$, $padding = 2$ and $bias = True$. There is one input channel, first layer outputs three channels and each other layer adds three more so that at the end there are twenty four channels. In between each layer there is LeakyReLU activation function and a 1D batch normalization.

The next important part of our network is the skip connections. The model training was difficult without them. When we run the same experiment mul-

CML 444 F1-score for different minimal rain intensities [I]							
$I \left[\frac{\text{mm}}{\text{hr}} \right]$	≥ 0	≥ 2	≥ 4	≥ 6	≥ 8	≥ 10	≥ 12
Model							
Thresholding	0.01	0.05	0.15	0.26	0.36	0.47	0.56
Shallow nn	0.41	0.72	0.76	0.80	0.82	0.82	0.81

Table 3.3: F1-score for CML 444 different models comparison**Figure 3.6:** Shallow neural network CML 56 ROC

multiple times, i.e. one CML as input and model training starting from random weights initialization, it either trained well or not at all. The reason was that the linear layers did not have any usable information when convolutional layers were not trained at the beginning. Then we introduced skip layers and the model started training consistently, i.e. multiple trainings under equivalent conditions ended up with similar results.

■ Fully connected layers

There are three fully connected layers with LeakyRelu in between of them and the input and output sizes are the same, only the output layer outputs two values which enter softmax function. Our experiments showed that there is a slight benefit in keeping input and output sizes the same compared to gradually decreasing them.

■ 3.4.1 Results

First of all, we want to prove that the idea of using CNN for the wet-dry classification can work. We will use the same CMLs as before. CMLs 56, 444

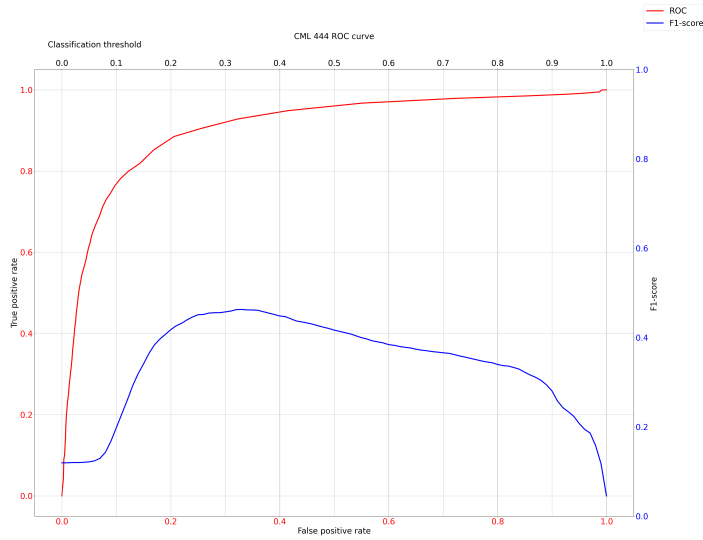


Figure 3.7: Shallow neural network CML 444 ROC

CML 56 F1-score for different minimal rain intensities							
Model \ $I \left[\frac{\text{mm}}{\text{hour}} \right]$	≥ 0	≥ 2	≥ 4	≥ 6	≥ 8	≥ 10	≥ 12
Thresholding	0.43	0.86	0.82	0.83	0.81	0.77	0.75
Shallow nn	0.62	0.95	0.94	0.94	0.94	0.94	0.95
Initial conv. nn	0.65	0.99	0.97	0.97	0.97	0.96	0.97

Table 3.4: F1-score for CML 56 different models comparison

and 547. We will use continuous data and classification reference C_{ref} 2.5. For each CML we will train a new model from beginning and we will use the same hyperparameters.

■ Training progress

Let's take a look at learning progress of CMLs 547 in figures 3.12, 56 3.11 and 444 3.13. We can see that the CML 444 was not able to train well. Learning curves for other two CMLs look standard. One reason could be that the model hyperparameters which work for CMLs 56 and 547 don't work for CML 444.

■ F1-score distribution

From the results in tables 3.4, 3.5 we can see that for I above $2 \frac{\text{mm}}{\text{hour}}$ the F1-score is around 0.94 for CML 444 and 0.98 for CML 56, which is better than previous results. When we include all I , the F1-score drops to around 0.6. The same trend for low I continues.

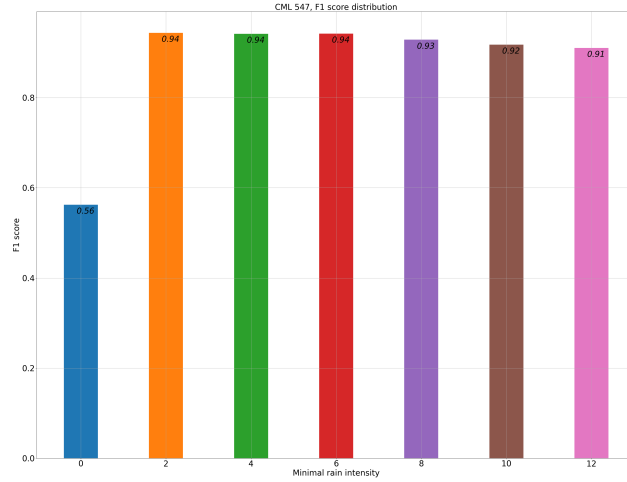


Figure 3.8: Shallow neural network CML 547 F1 test scores

CML 444 F1-score for different minimal rain intensities [I]							
I [$\frac{\text{mm}}{\text{hour}}$]	0	2	4	6	8	10	12
Model							
Thresholding	0.01	0.05	0.15	0.26	0.36	0.47	0.56
Shallow nn	0.41	0.72	0.76	0.80	0.82	0.82	0.81
Initial conv. nn	0.17	0.95	0.94	0.95	0.95	0.94	0.93

Table 3.5: F1-score for CML 444 different models comparison

■ Misclassifications

In figure 3.15 we can see that the model disregards small attenuation increase caused by a $2 \frac{\text{mm}}{\text{hour}}$ rain and classifies this data as no rain. The reason for this is shown on the next figure 3.16. There is also a small attenuation increase, but this time it is classified as rain, but the reference says no rain even though we can see that most likely it was raining somewhere on the CML path. So this error is caused by the reference inaccuracy where the same or very similar situation is rated sometimes as rain and other times as no rain. It is then impossible for the network to distinguish between these two cases.

The same situation can be seen for the CML 444 in figure 3.17 as well, there is an attenuation increase presumably due to rain, but the reference tells otherwise. I have created multiple graphs depicting misclassification situations and all agree that most errors happen in situations with mild rain around $2 \frac{\text{mm}}{\text{hour}}$ and they are caused by the reference not being accurate enough.

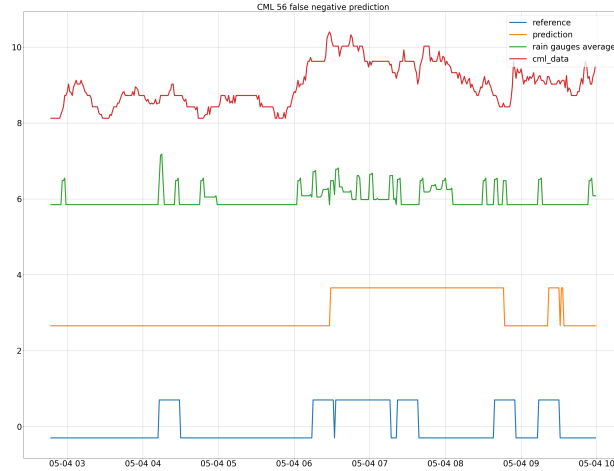


Figure 3.9: Shallow nn misclassification for low rain intensity

ROC

We can take a look at ROCs. For CML 56 in figure 3.19 we can see a good model performance for $C_{thres} = 0.18$ for which $FPR = 0.012$. Another observation is that the model is quite accurate. Even for high C_{thres} the maximal $FPR = 0.07$. The same figure for CML 444 in figure 3.18 shows that our model has difficulty to reach similar performance for this particular CML.

3.4.2 Discussion

The CNN shows promising results in wet-dry classification. Its performance beats all previous models. The main source of error comes from situations with low rain intensity. We will try different versions of this model and compare which one works best and whether we can improve predictions for low I .

3.5 BCM further experiments

In the next part, we will do experiments in which the BCM in figure 3.10 stays the same and we will vary input data. In following paragraphs we will describe all other classification experiments.

We summarize results for CML 56 3.1 and 444 3.1 in two tables. In our experiments we will vary the input data sequence length (**SL**) measured in days, continuous data flag (**CD**) and A_{res}

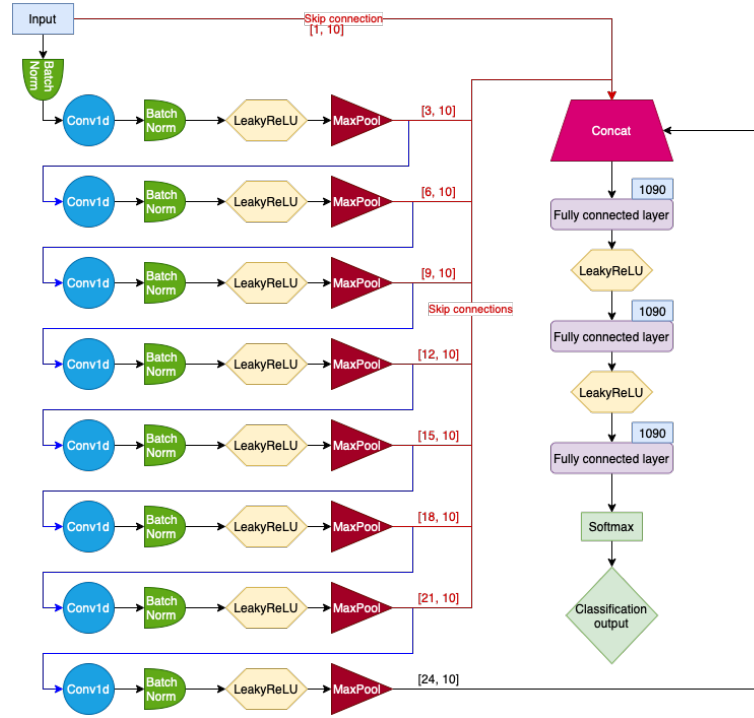


Figure 3.10: Conv net classification

3.5.1 Training details

CML 56

Now, we will take a closer look at training results for CML 56. The best result is reached with $SL = 1\text{day}$, $CD = \text{True}$ and $A_{res} = 1\text{min}$. We can see in the table 3.1 that the results are quite even. Training progress figure 3.20 shows a standard evolution. The network has trained for 100 epochs and we observe that if the training had continued longer a slightly better performance might have been reached. Based on this we conclude that training parameters were chosen well. Hyperparameters are learning rate: $lr = 3 \cdot 10^{-8}$, batch size: $b = 200$ and Cross entropy weights: $(w_0, w_1) = (0.1, 0.9)$. The weight w_0 scales the loss for $C_{ref} = 0$, i.e. no rain and w_1 scales the loss for rain events.

We can say that the worst performance is for $SL = 1\text{day}$, $CD = \text{False}$ and $A_{res} = 1\text{min}$, because it does not reach best result in any category. Its training progress in figure 3.21 shows that the validation loss does not evolve smoothly. Hyperparameters stayed the same as before. The worst performance of the F1-score mirrors the worst training progress of the validation loss.

When we take a look at the ROC curve with F1-score evolution based on C_{thres} for the worst result in figure 3.22 we can see that we could get a better F1-score by setting $C_{thres} = 0.3$. In comparison with the same figure for the best result 3.23a we get more FPs as the C_{thres} increases. For $C_{thres} = 0.7$, $FPR^{best} = 0.062$ and $FPR^{worst} = 0.090$. Another difference

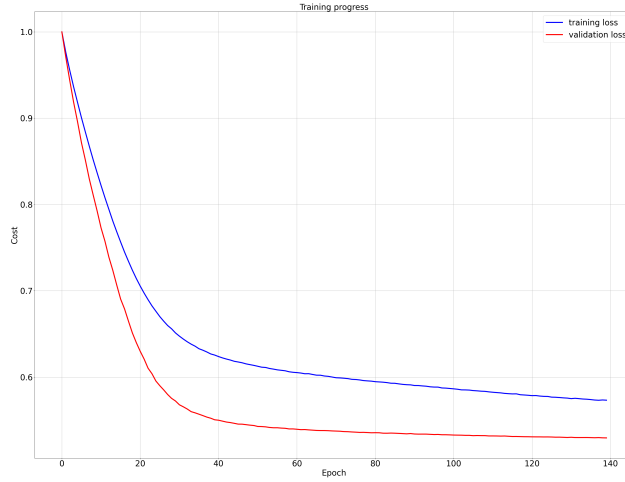


Figure 3.11: Conv. net initial experiment training progress CML 56

between results is that F1-score is increasing in figure 3.23a but after initial rise it is decreasing in figure 3.22.

■ CML 444

CML 444 results 3.6 are overall worst across all I ranges. For $I \geq 0 \frac{\text{mm}}{\text{hr}}$ is the F1-score 0.3 points lower then in case of CML 56. The best result is reached with parameters $SL = 1\text{day}$, $CD = \text{True}$ and $A_{res} = 1\text{min}$. Those are the same parameters as for CML 56. The worst result was achieved with same parameters as for CML 56, $SL = 1\text{day}$, $CD = \text{False}$ and $A_{res} = 1\text{min}$

When we compare training progress plots we can see that the one belonging to the best result, figure 3.26, looks like the one from our initial classification experiment, figure 3.13. The validation loss decreases for 40 epochs and then starts increasing. Interestingly, the second training progress depicted in figure 3.27 of the worst result looks like what we would expect.

We can say that the worst performance is for $SL = 1\text{day}$, $CD = \text{False}$ and $A_{res} = 1\text{min}$, because it does not reach a best result in any category. Its training progress in figure 3.21 shows that the validation loss does not evolve smoothly. Hyperparameters stayed the same as before. The worst performance of the F1-score mirrors in the worst training progress of the validation loss.

Next, we take a look at the ROC curve with F1-score evolution in figures 3.28, 3.29. They show that for CML 444 there is a much higher FPR. For $FPR^{444} = 0.5$ and $FPR^{56} = 0.05$ for $C_{thres} = 0.5$. Predictions with high certainty produce a lot of FPs.

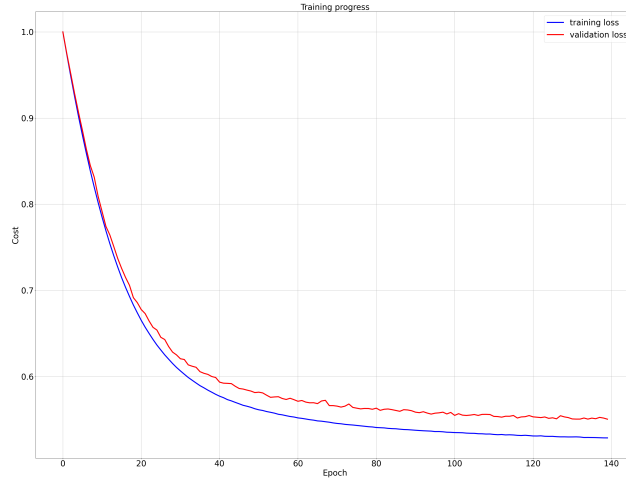


Figure 3.12: Conv. net initial experiment training progress CML 547

			minimal rain intensity [mm/hr]						
SL	CD	A_{res}	0	2	4	6	8	10	12
1 day	False	1	0.31	0.80	0.81	0.80	0.78	0.75	0.72
1 day	True	1	0.21	0.96	0.96	0.97	0.96	0.95	0.95
2 days	False	1	0.54	0.94	0.94	0.95	0.95	0.95	0.95
2 days	True	1	0.17	0.95	0.94	0.95	0.95	0.94	0.93
1 day	False	2	0.42	0.93	0.94	0.95	0.95	0.94	0.95

Table 3.6: CML 444 F1-scores for different classification experiments

3.5.2 Misclassification visualization

CML 56

Next interesting aspect to demonstrate is where predictions are wrong. We have plotted predictions, reference, RGs average and A for misclassified samples. Figure 3.24 depicts FP and FN predictions for low I . We can see that it portrays a situation, where on 9. June at 00:15 o'clock and at 01:45 o'clock the same $A = A_b + 1.7\text{dB}$ results in different C_{ref} .

Next figure 3.25 shows misclassification for high RG average. We observe a medium rain which is mostly not detected by the CML 56, it's attenuation raises insignificantly for the most part.

CML 444

When we depict wrongly classified situations we can observe how the model uncertainty manifest itself. Following figure 3.30 presents situation from 17. September around 18:30 o'clock. There are rapidly changing predictions after

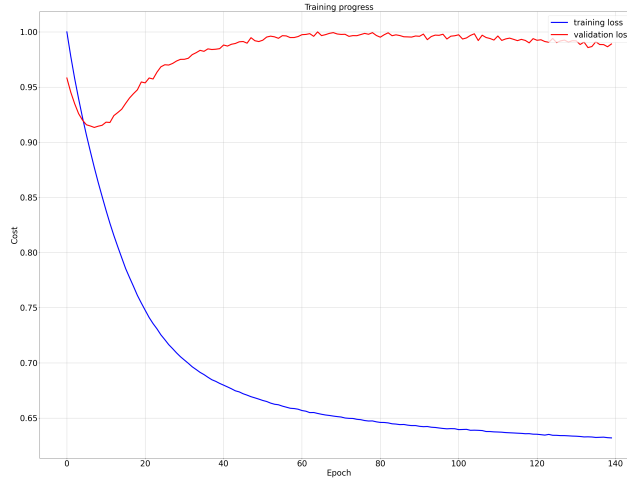


Figure 3.13: Conv. net initial experiment training progress CML 444

a medium rain has stopped. This is interesting, because $A = A_b$ from before the rain.

Second wrongly classified situation is analogues to those we have seen before. There is a $2 \frac{\text{mm}}{\text{hr}}$ to $4 \frac{\text{mm}}{\text{hr}}$ rain which changes A^{444} insignificantly. Consequently the model predicts no-rain.

Another figure 3.31 shows a very difficult to evaluate situation. There is an approximately half hour long mild to medium rain which does not reflect in the A^{444} . This event is part of a 6 hour time window in which A^{444} is volatile. As result, predictions oscillate between rain and no-rain as the model is not sure what to predict.

3.5.3 Results

We present the final wet-dry classification results for CML 56 3.7 and CML 444 3.8.

Experiments show that $SL = 1\text{day}$ is better then $SL = 2\text{days}$ and data for which time continuity is ensured work better then without it. Lastly, $A_{res} = 2$ makes results worst.

Lastly, we want to compare our best performing model and compare it with the Polz et al. version. We can compare two metrics. The ROC and MCC. The ROC in figure 3.23a shows $\text{TPR} = 0.92$ and $\text{FPR} = 0.045$ for $C_{thres} = 0.5$. The Polz et al. version achieved $\text{TPR} = 0.78$ and $\text{FPR} = 0.045$. Our version performed slightly better in terms of TPR/FPR ration

The MCC metric for our best performing classification model trained on CML 56 compared to the Polz et al. version is shown in figure ???. We can see that our results are very close to each other. The best MCC score for all rain intensities achieved by Polz et al. was 0.69. Our experiments confirm

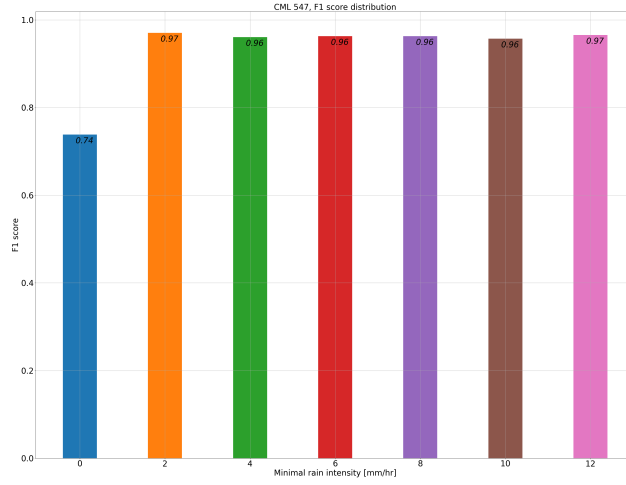


Figure 3.14: CML 547 F1-score distribution

CML 56 F1-score for different minimal rain intensities							
Model \ $I \left[\frac{\text{mm}}{\text{hr}} \right]$	0	2	4	6	8	10	12
Thresholding	0.43	0.86	0.82	0.83	0.81	0.77	0.75
Shallow nn	0.62	0.95	0.94	0.94	0.94	0.94	0.95
BCM initial	0.65	0.99	0.97	0.97	0.97	0.96	0.97
BCM best	0.65	0.99	0.98	0.98	0.98	0.98	0.97

Table 3.7: Final F1-score for CML 56 different models comparison

this performance. Our best model was able to reach $MCC = 0.71$ and it gets above 0.8 for higher rain intensities.

3.5.4 Discussion

Wet-dry classification results show that it can be done by a CNN well. The only problem arises for low I . Misclassification visualizations have shown that it is not a problem of the network but rather a question how to define the problem better. It is problematic to train a model on a single CML with reference that does not match reality for some data. Moreover, the reference is inconsistently wrong, that means that two identical situations are evaluated differently. One possible solution might be to train a model on multiple CMLs and generalize the reference for some area. This approach will be used in one of our regression models.

Next observation is that one method has significantly different results for different CMLs. Model trained on CML 56 performed much better than the same model trained on CML 444. It is not surprising, because earlier figures

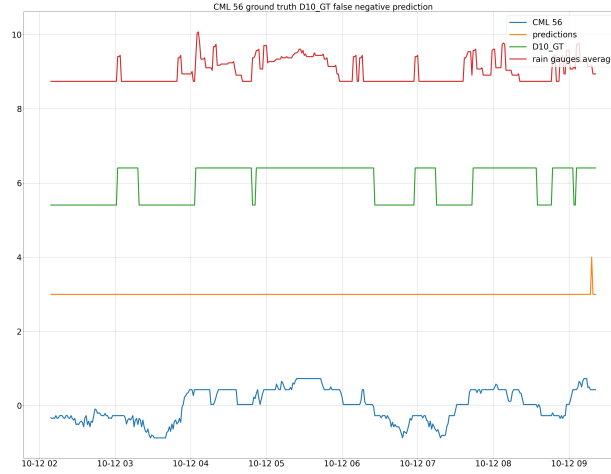


Figure 3.15: CML 56 misclassifications

CML 444 F1-score for different minimal rain intensities [I]							
Model \ I [$\frac{\text{mm}}{\text{hr}}$]	0	2	4	6	8	10	12
Thresholding	0.01	0.05	0.15	0.26	0.36	0.47	0.56
Shallow nn	0.41	0.72	0.76	0.80	0.82	0.82	0.81
BCM initial	0.17	0.95	0.94	0.95	0.95	0.94	0.93
BCM best	0.21	0.96	0.96	0.97	0.96	0.95	0.95

Table 3.8: Final F1-score for CML 444 different models comparison

1.8 and 1.6 showed how different in terms of A_b behavior different CMLs are.

Another fact we learned is that the Shallow neural network model performed quite well. It showed a good ability to generalize. It had the highest F1-score $F_1^{444} = 0.41$ for $I > 0 \frac{\text{mm}}{\text{hr}}$ and showed competitive results for CML 56 as well. The convolutional model displayed inability to generalize well for CML 444 in comparison. It was able to overfit on the training data but its performance did not carry over to validation data.

Lastly we compared our model performance to the Polz et al. version and we were able to achieve comparable results. The Polz version trained the model on 800 CMLs located in Germany and they had five months worth of data for each CML. Their model did not utilize any CML attributes to account for different CML lengths. We trained our model on one CML with data gathered during three years. Even though our approaches differ in multiple parameters our results are very similar.

3. Wet-dry classification

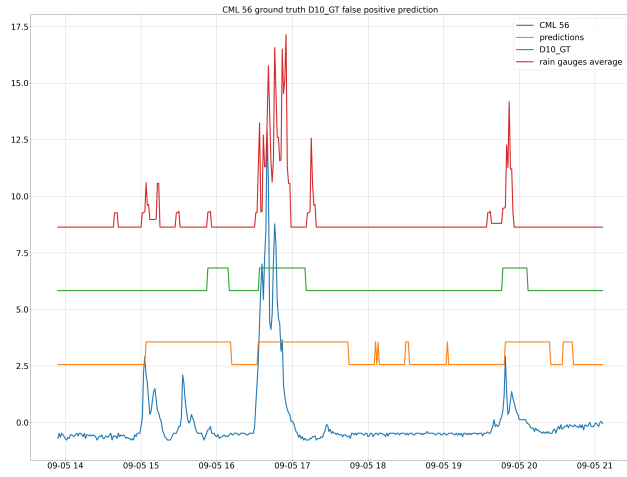


Figure 3.16: CML 56 misclassifications

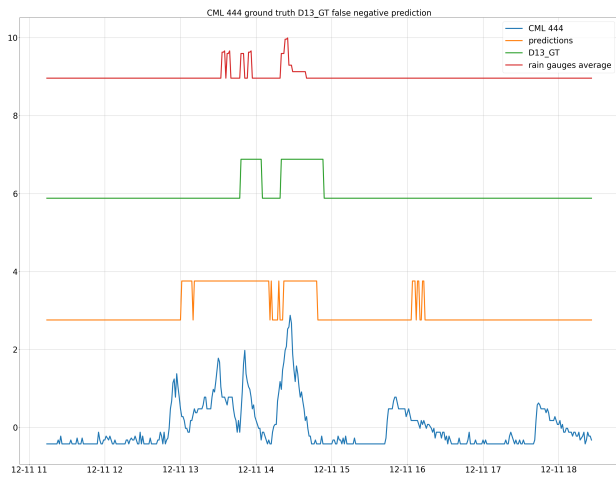


Figure 3.17: CML 444 misclassifications

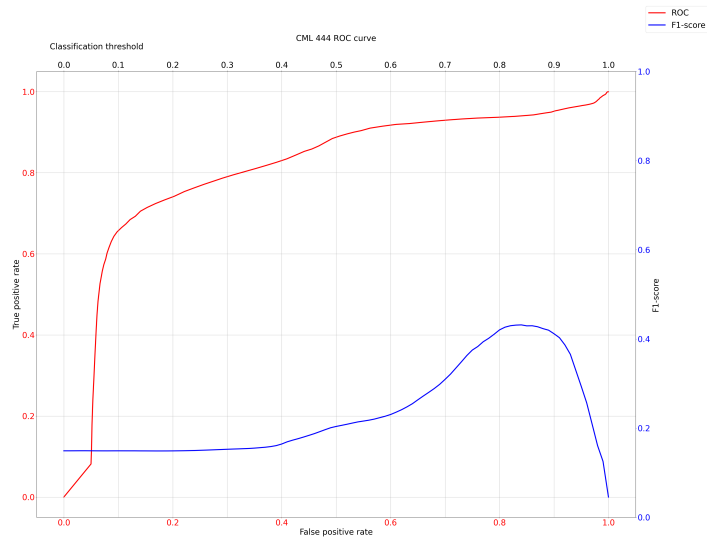


Figure 3.18: CML 444 initial conv. net continuous data

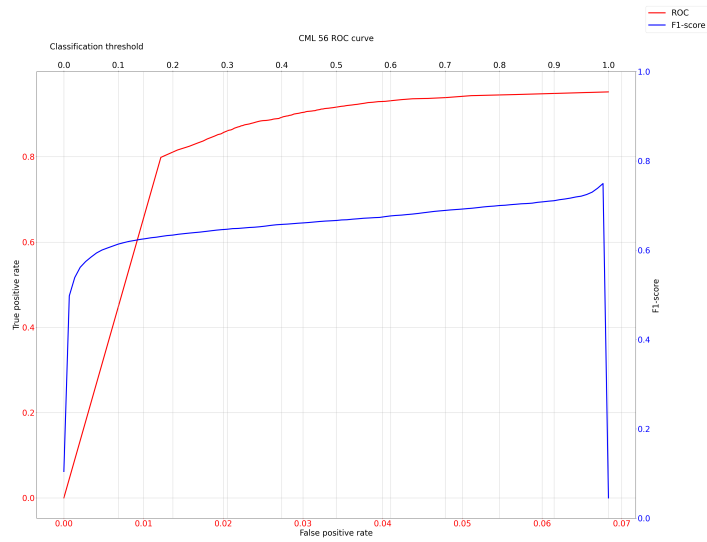


Figure 3.19: CML 56 initial conv. net continuous data

3. Wet-dry classification

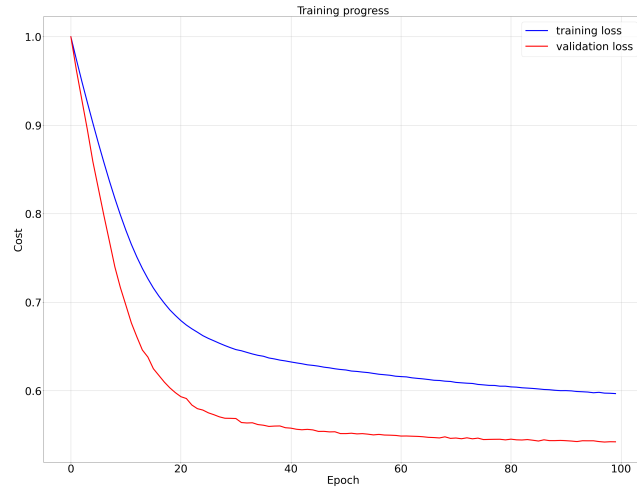


Figure 3.20: CML 56 classification training progress SL = 1day, CD = True

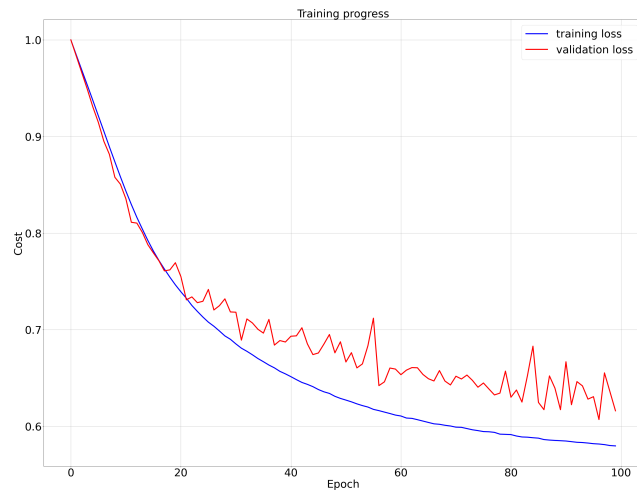


Figure 3.21: CML 56 classification training progress SL = 1day, CD = False

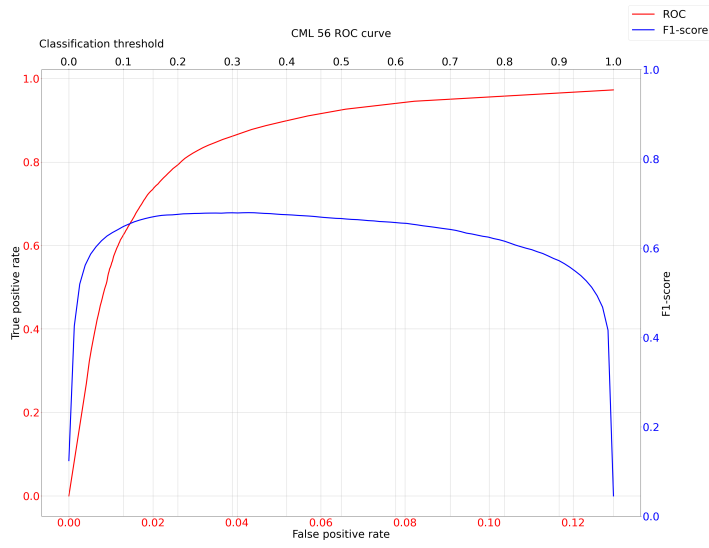
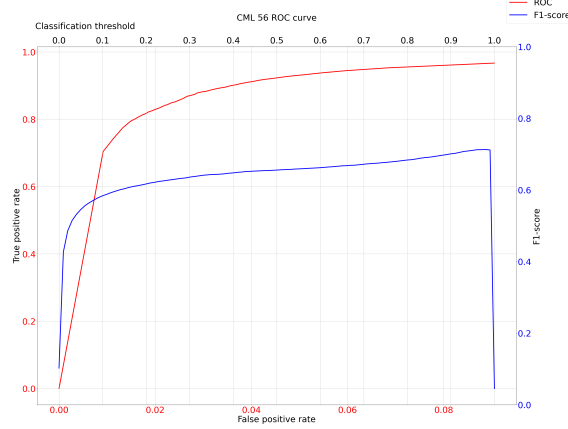
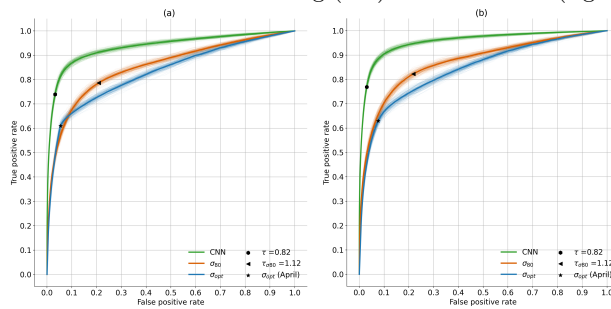


Figure 3.22: CML 56 ROC with F1-score evolution worst result

(a) : CML 56 ROC with F1-score evolution best result



(b) : Polz et al. ROC for training (left) and validation (right) data



3. Wet-dry classification

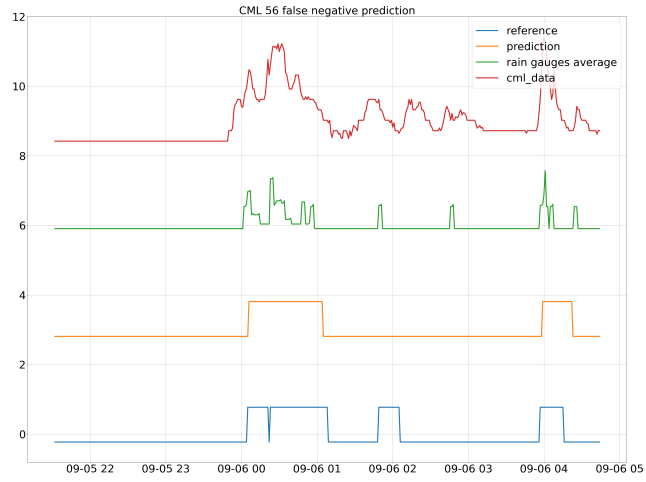


Figure 3.24: CML 56 wrong predictions mild rain

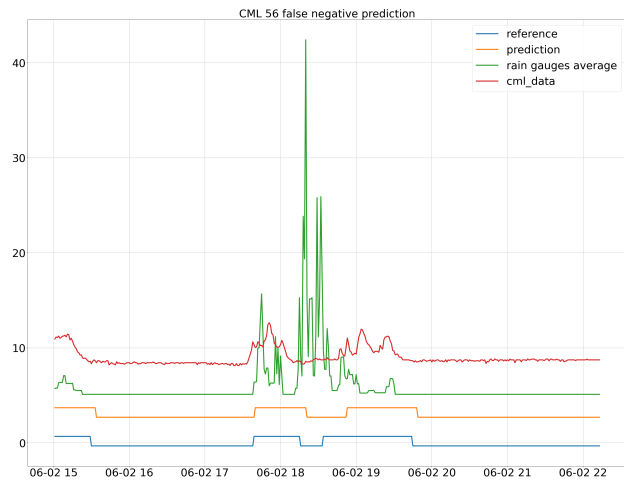


Figure 3.25: CML 56 wrong predictions medium rain

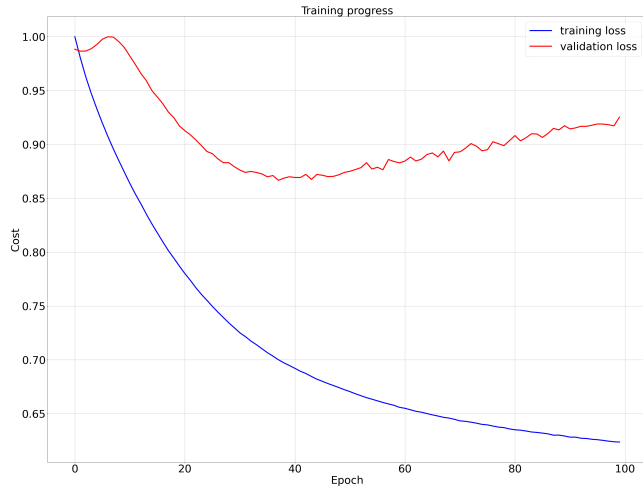


Figure 3.26: CML 444 classification training progress SL = 1day, CD = True

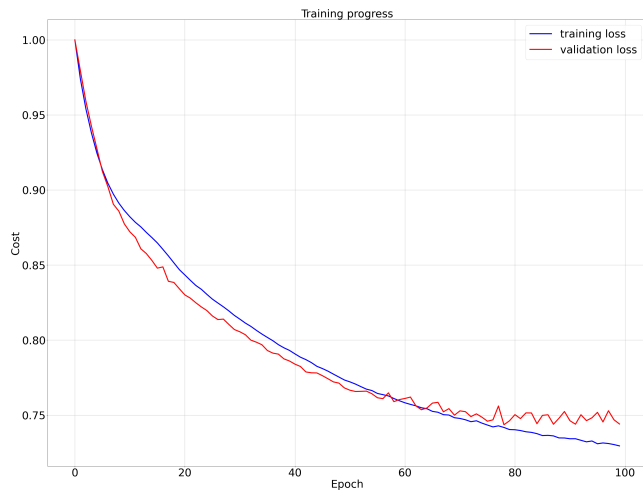


Figure 3.27: CML 444 classification training progress SL = 1day, CD = False

3. Wet-dry classification

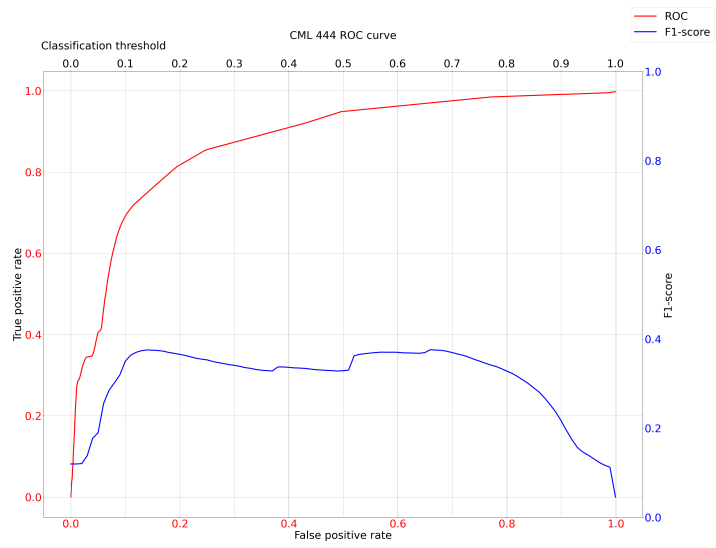


Figure 3.28: CML 4 ROC with F1-score evolution worst result

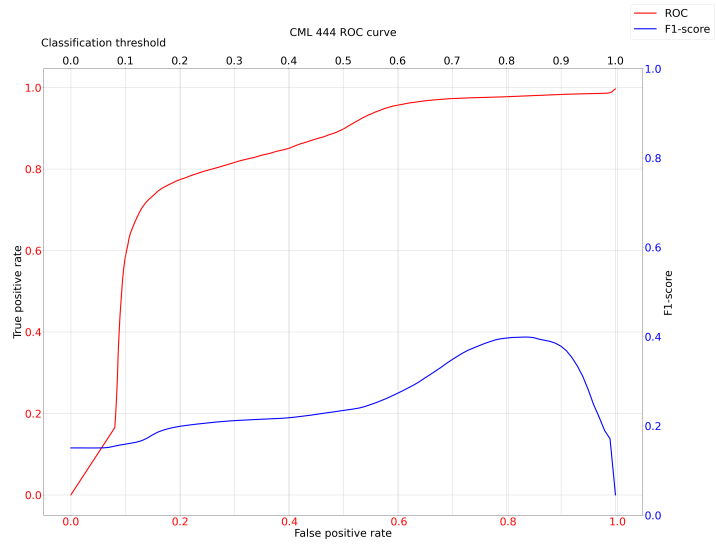


Figure 3.29: CML 444 ROC with F1-score evolution best result

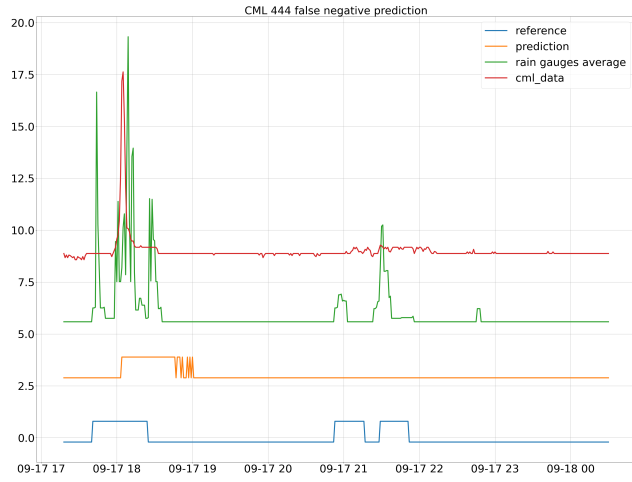


Figure 3.30: CML 444 wrong predictions mild rain

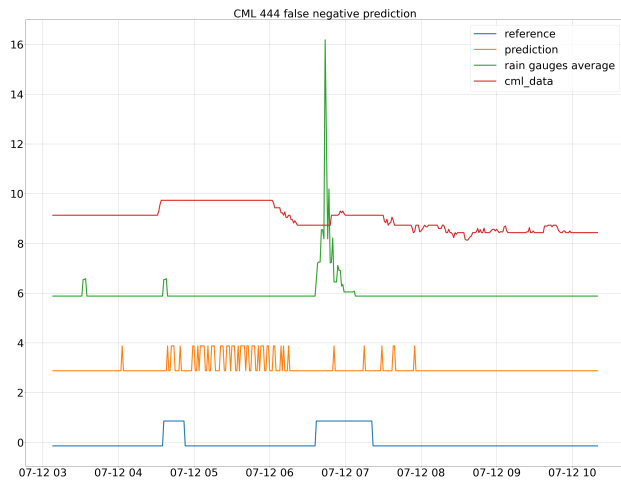
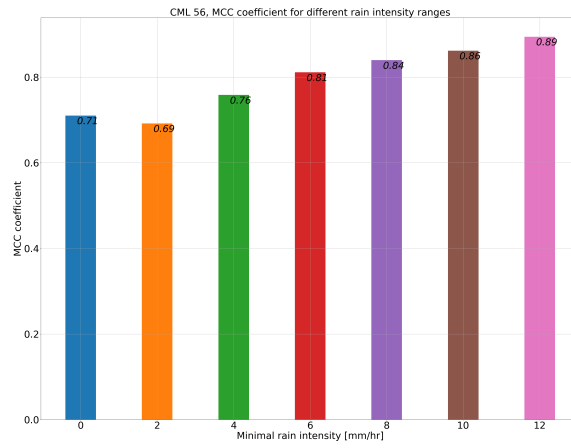


Figure 3.31: CML 444 wrong predictions medium rain

3. Wet-dry classification

(a) : CML 56 MCC for different rain intensities ranges



(b) : Polz et al. [16] MCC results

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

Chapter 4

Rain intensity estimation

The next task is to develop a model that predicts I . As reference the R_{ref}^d 2.6 is used. Because the convolutional model worked well on wet-dry classification task, we will adapt it for rain estimation

The estimation of rain intensity using CMLs is an opportunistic method for precipitation monitoring.

4.1 Architectures

At first, we used the same convolutional network with 24 output layers as before 4.1. We call it the **Basic regression model**. We trained it on attenuation data from single CML, then on multiple CMLs. Results were really bad 4.1 so we decided to add more channels to the convolutional layer. Our inspiration was the VGG-16 architecture. This improved version is called the **Improved regression model (IRM)** and its architecture diagram is shown here 4.2.

The last model we created is the **Multi-channel input regression model (MCIRM)** 4.3. We thought that if one R_{ref} gives inaccurate reference for one CML then we can create an input \mathbf{x} with multiple channels and each channel will contain different attenuation data for the same time instance t . The newly designed input looks like this 4.4.

4.1.1 The Improved regression model

The IRM uses 1440 samples on input. They are normalized by a 1D batch normalization. After that there are eight convolutional layers each constitutes of a 1D convolution followed by a 1D batch normalization, leaky ReLU activation function and a max pooling layer which decreases the number of samples in each channel by half. All convolution outputs go into a shallow linear network made of three fully connected layers with a leaky ReLU in between. This network outputs one value which represents the rain intensity in $\frac{\text{mm}}{\text{hr}}$.

The 1D convolution has `kernel_size = 5` `stride = 1` `padding = 2` `bias = False`. We use the leaky ReLU activation function to prevent vanishing gradient problem. Another important part, again, are the skip-connections.

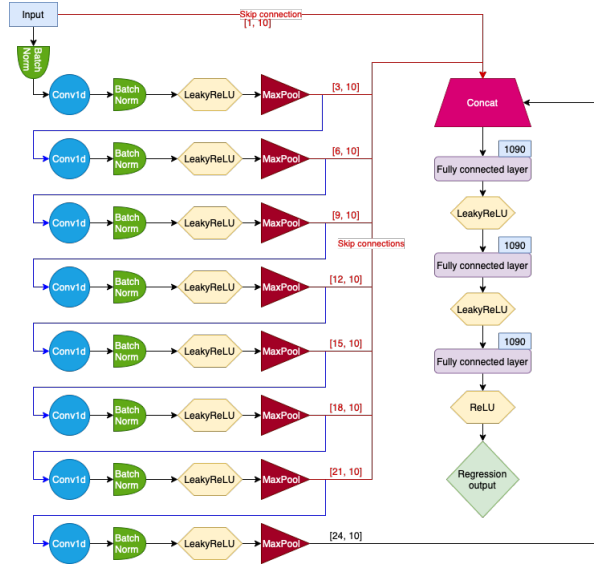


Figure 4.1: Basic regression model architecture

Output of the convolutions has 512 channels, each containing five values, and seven skip-connection outputs. Every skip-connections contains ten values in to the 1D convolution layer corresponding number of channels. This should ensure that the next part, the shallow linear layer, receives relevant features from the training beginning. The output is fed in a shallow linear network consisting of three fully connected layers and a LeakyReLU activation function between them. We also placed a ReLU function after the output layer to make sure that the output is never negative as it has no physical interpretation.

Intended downside of the model is that it must be trained and works only with one CML. There is no information about CML attributes going into the model and thus it cannot generalize to unseen CMLs. This restriction means that the training data size is also limited to the data size of one CML which is around 750 million samples. We build this model to prove that such architecture can show promising results and once we found that it has, we upgraded this architecture into the *Multi-channel input regression model*. Complete results overview is located in the section *Results*.

4.1.2 Multi-channel input regression model

Once the IRM showed good results, its error was lower then GRU for all intensities other then $1 \frac{\text{mm}}{\text{hr}} < I \leq 5 \frac{\text{mm}}{\text{hr}}$, we extended the IRM so that it can work with multiple CMLs. Our goal is to find one on CML independent model.

The MCIRM architecture differs from IRM in three aspects. The IRM input consists of 1444 samples from one CML. In case of MCIRM the input consists of n channels. In each channel there are 1444 samples from a different CML. For a rain estimation for time t we have n channels CH containing

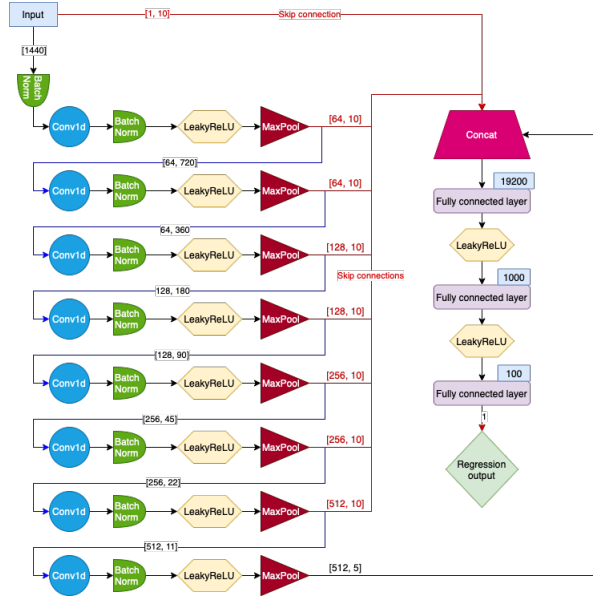


Figure 4.2: Improved regression model architecture inspired by VGG-16

data like this 4.1

$$CH^n(t) = \{A^n(t), A^n(t-1), \dots, A^n(t-1439)\} \quad (4.1)$$

In this way the model can work with attenuation data from multiple CMLs from the same time interval. This gives the model a broader information about the weather situation in an area. Moreover, the shallow linear layer in the IRM can now learn the importance of each CML feature. For example if there is a CML with a short length, then the A_r consists mostly of A_{waa} and the model could use it to estimate A_{waa} for all CMLs. In other words, the shallow linear network located at the end of the IRM has more diverse information to work with and can learn to combine different CML features.

The second difference is that the IRM part of the model outputs not one but n values $\mathbf{y}_{tmp} = (y_1, y_2, \dots, y_n)$. Each value is thought of as A_r^c . The job of the adjusted IRM model now is to estimate the rain induced attenuation for each CML. This output is then processed by a newly developed division layer.

The division layer's job is to calculate rain intensities for each RG. 4.3. We know that $A_r = l \cdot I \cdot c$, where c is independent constant. So we ask the model to find A_r and then the division layer calculates I as follows 4.2. The model weights correspond to $\frac{s}{c}$ where s is a scaling factor which determines the CML relevance for a given RG. So if the network figures out that a particular CML has poor correlation with a RG, it can scale its importance down.

$$I = \frac{A_r}{lc} \quad (4.2)$$

$$w_{ij} = \frac{1}{c} s_{ij}$$

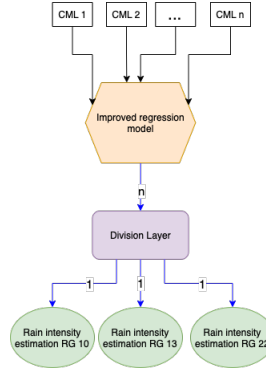


Figure 4.3: Multi-channel input model

$$\mathbf{Y}_{rain_estimate} = \mathbf{Y}_{IRM_out} \begin{bmatrix} \frac{w_{11}}{l_{CML1}} & \frac{w_{12}}{l_{CML1}} & \frac{w_{13}}{l_{CML1}} \\ \frac{w_{21}}{l_{CML2}} & \frac{w_{22}}{l_{CML2}} & \frac{w_{23}}{l_{CML2}} \\ \vdots & \vdots & \vdots \\ \frac{w_{n1}}{l_{CMLn}} & \frac{w_{n2}}{l_{CMLn}} & \frac{w_{n3}}{l_{CMLn}} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (4.3)$$

The division layer has three outputs, one for each RG. We experimented with three and nine CML inputs and we chose them in such a way that there is one and three CMLs near each RG. Also, we wanted CMLs with different lengths, therefore the shortest CML has $l^{333} = 186\text{m}$ and the longest CML has $l^{62} = 5.795\text{m}$. For the purpose of model performance evaluation we calculated for each time instance the average of R_{ref} and the MCIRM output average and computed same metrics as for all other models.

$$x(t) = \begin{bmatrix} A^{CML1}(t) \\ A^{CML2}(t) \\ \vdots \\ A^{CMLn}(t) \end{bmatrix} \quad (4.4)$$

4.2 Loss functions

Another important aspect of a model is its loss function. We have two goals in mind. For one, whether a model's prediction error is $2 \frac{\text{mm}}{\text{hr}}$ for $R_{ref} = 1 \frac{\text{mm}}{\text{hr}}$ or $R_{ref} = 40 \frac{\text{mm}}{\text{hr}}$, is a big difference. In the former case it is a huge error, but in the latter it is a very accurate prediction. Therefore the first loss we use is the **Huber loss** 4.6 with $\delta = 1$. As result, all rain estimate errors for $R_{ref} \leq 1 \frac{\text{mm}}{\text{hr}}$ are squared and every other estimate errors are linear.

Another attribute of our data set is the data imbalance. In order to make the model focus more on rain events rather than on dry ones we adopted the **scaled MSE loss**, [9] 4.5 with $\delta_s = 0.95$ and $\delta_r = 5$. This loss ensures that a prediction for larger $R_{ref} > 0$ is more important than for $R_{ref} = 0$. Experiments showed that the scaled MSE loss works better with MCIRM

Network attributes				Rain intensity range [mm/hr]					
Architecture	Loss func.	CD	#CMLs	$0 < y^r \leq 1$	$1 < y^r \leq 5$	$5 < y^r \leq 15$	$15 < y^r \leq 25$	$25 < y^r \leq 35$	$35 < y^r \leq 100$
TSN [9]	scaled MSE	True	40	1.6	0.65	0.55	0.52	0.80	0.60
Basic	scaled MSE	True	1	10.13	1.06	1.06	1.01	1.01	1.07
Improved	scaled MSE	True	1	2.45	1.37	0.69	0.34	0.41	0.5
Improved	Huber	True	1	0.99	0.72	0.53	0.49	0.49	0.6
Multi- channel	Huber	False	3	0.95	0.84	0.68	0.75	0.75	0.85
Multi- channel	scaled MSE	False	3	0.98	0.91	0.65	0.58	0.47	0.51
Multi- channel	scaled MSE	False	9	1.31	0.98	0.57	0.63	0.58	0.70

Table 4.1: NRMSE regression experiments comparison

and in case of the IRM the Huber loss performed better for $R_{ref} \leq 15 \frac{\text{mm}}{\text{hr}}$ and the scaled MSE for $R_{ref} > 15 \frac{\text{mm}}{\text{hr}}$.

$$\begin{aligned}
 J_{MSE}(\mathbf{x}, \mathbf{y}) &= \sum_{i=0}^n (x_i - y_i)^2 \\
 RDF(y_{n,i}^r) &= 1 - \gamma_s \exp(-\gamma_r \cdot y_{n,i}^r) \\
 J_{reg}(y_{n,i}^r, \hat{y}_{n,i}^r) &= RDF(x_{n,i}^r, \gamma_s, \gamma_r) \cdot J_{MSE}(y_{n,i}^r, \hat{y}_{n,i}^r)
 \end{aligned} \tag{4.5}$$

where RDF is the scaling factor of the estimation loss and $\gamma_s \in [0, 1]$ and $\gamma_r > 0$ are hyperparameters.

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta, \\ \delta(|y - \hat{y}| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases} \tag{4.6}$$

4.3 Results

All regression models were trained on 250+ epochs with $lr = 3 \cdot 10^{-8}$. The lr was never changed during the training. Validation loss stopped decreasing around epoch 200 for all models. One example of training progress of the Multi-channel scaled MSE model is shown here in figure 4.4

All interesting experiments results are shown in 4.1. In the first line there is the TSN model [9] which is our benchmark. The main differences between our models and TSN model are that we only used 1, 3 and 9 CMLs for input, TSN used 40 CMLs and that TSN uses multiple CML attributes as model input whereas we only use the CML lengths in the division layer.

We can see that our experiments match the TSN results. Our best model using single CML is the Improved model with Huber loss function and continuous data. It shows better performance than TSN for all I ranges other than $1 < I \leq 5$. We observe that the Huber loss compared to the scaled MSE loss shifted model training focus from high I to lower I , but managed to have competitive performance for higher I as well. We can look at its results from a perspective of a box plot in figure 4.5. It shows that the mean RMSE lies above median and for $I \leq 5$ it is located above the 75% threshold.

Interesting observation is that the more CMLs we included into training, the worse the results. The reason why this is the case for our Multi-channel model might be that the data size got reduced by approx. 25%, because for

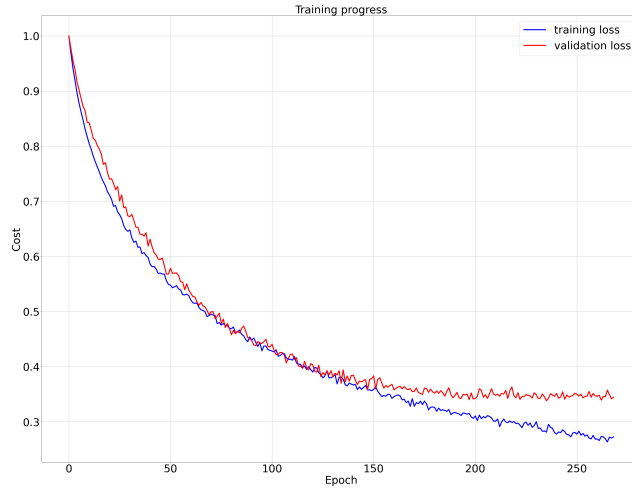


Figure 4.4: Multi-channel scaled MSE loss regression models training progress

the input we needed valid data from multiple CMLs from the same time. The average training data size for one CML is 750 000 samples, the Multi-channel input had only 568 000 samples.

For comparison, we provide the same plot for the Improved model with scaled MSE loss figure 4.6. We can see that their performance is quite similar and one could tweak the scaled MSE and Huber loss hyperparameters to shift models training focus on different I segments.

The Multi-channel models performed both well but based on their box plots the one with scaled MSE loss in figure 4.7 did better then the second one in figure 4.8.

4.3.1 Discussion

We have shown that a convolutional layer based model is capable of Wet-dry classification and Rain intensity estimation. Our Regression models were able to reach similar results to the TSN model. The key difference between our models and TSN being that we did not use as many CMLs.

Advantage of our approach are that the convolutional network does not use many parameters and is relatively easy to train, at leased compared to regression models.

Figures depicting misclassifications and data exploration showed that moving forward we should think more about the problem definition with respect to references we have. As it is right now, we don't have accurate references and as capable as neural networks can be, they cannot deal with it well. Their correlation is not good enough. This results in contradictory reference with respect to the attenuation, thus hindering the model training. Therefore we should find a better problem definition.

My proposal in moving forward is to prepare a dedicated experiment

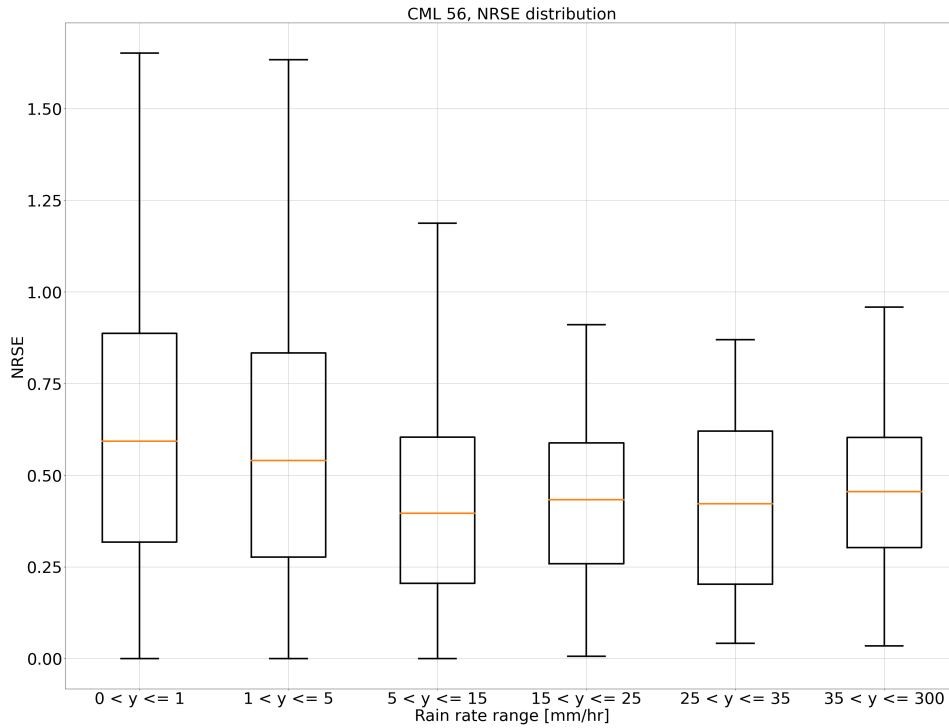


Figure 4.5: Improved regression model using Huber loss

where RGs are placed directly underneath CML paths. Then, with accurate reference data, it should be much easier to train a general rain estimating model. Once the model would be trained, its performance could be validated on so many already existing CML signal datasets. Finding good rain estimation model is valuable so it would be worth the money investment.

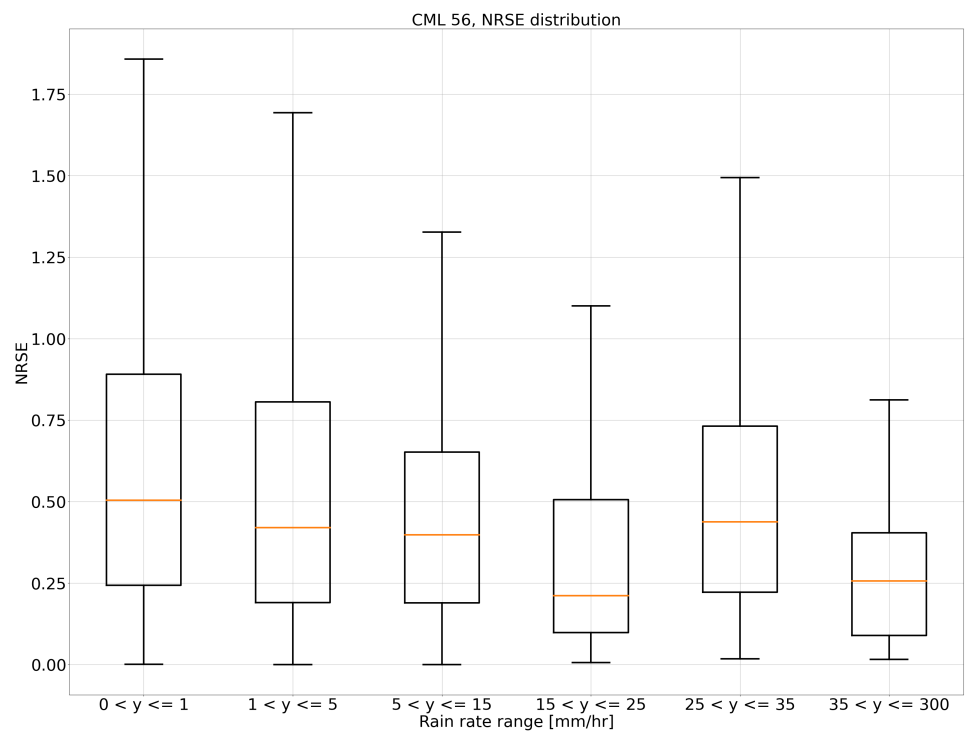


Figure 4.6: Improved regression model using Huber loss

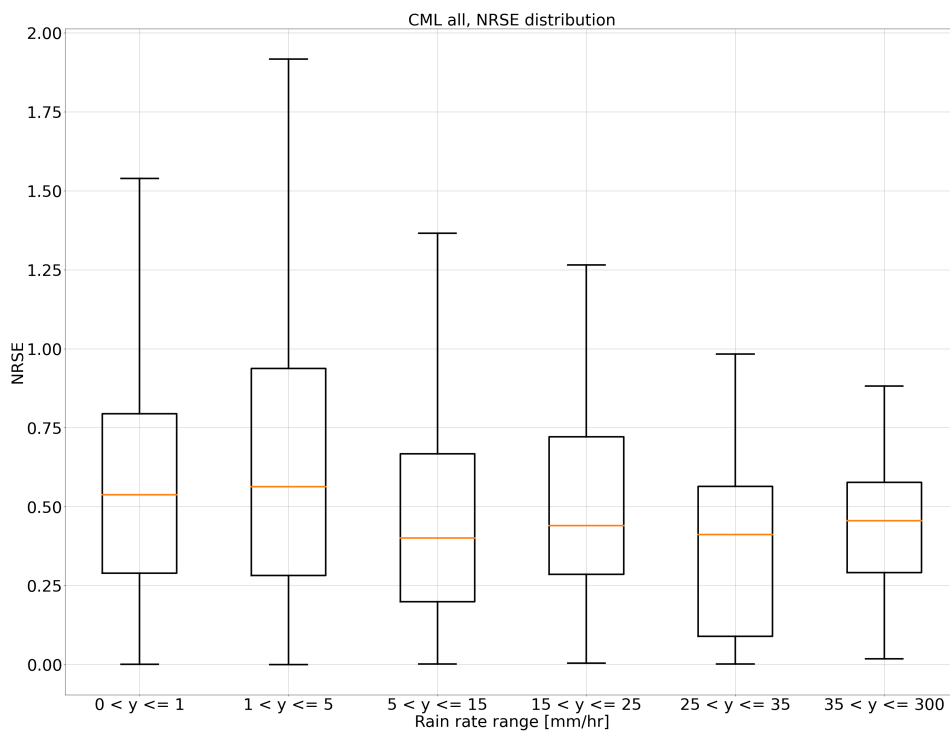


Figure 4.7: Multi-channel regression model using scaled MSE loss

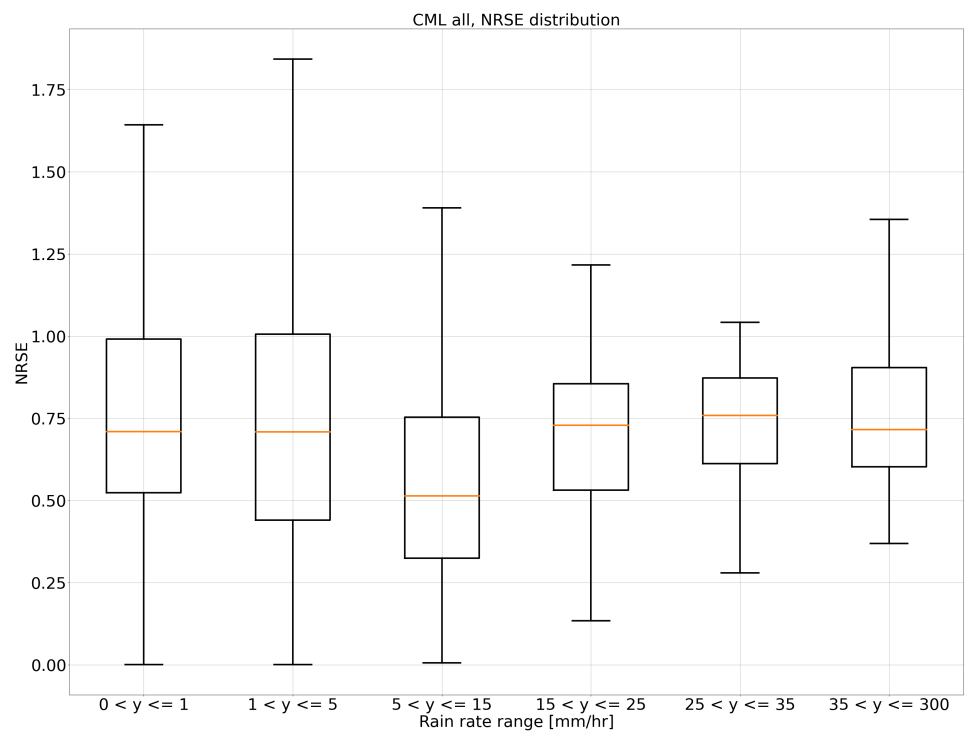


Figure 4.8: Multi-channel regression model using scaled Huber loss

Chapter 5

Software solution

We have developed a software solution for data preparation, model training and results visualization. It is located on /datagrid (must be specified). It was created with the idea in mind that it should be useful by the future researchers, who will wish to continue on the rain estimation from CML attenuation topic. There are 3 standalone scripts for data pre-processing, one module for model training and one for automatic result visualization and evaluation.

■ Attenuation calculation

The scripts name, which prepares the attenuation data, is *01_attenuation_calculation.py*. In order to make it work, valid paths must be filled in in the *main()* function. The *folder_containing_cml_data* variable points to the CML data parent directory, *path_to_save_data* variable points to the folder where results will be stored and *path_to_temperature* variable points to *temperature_Prosek.csv* file in which temperatures are stored.

There is a parameter *sample_freq = 1*, which is set to 1 minute. It defines the output data time resolution. We used 1 minute but other time resolutions can be easily generated and experimented with.

If you wish to prepare your data in a different way, you can do it. It will be perfectly usable with the rest of the program as we will describe later in this section. There is no unified output data format for this step. The unification process takes place in the model training module.

The output of this script is conforms to 2.4 and additionally any S_j^c $||S_j^c| < 240$ is discarded. It is generated into the folder defined by *path_to_save_data* variable and consists of one folder for each CML. In each folder there are multiple .pkl files. Each file contains continuous data. There are 100 gaps on average, therefore around 100 files are generated for each CML. This ensures that data is continuous but it's average cost is the loss of $100 * n$ data where n is the sequence length needed for a model input.

■ CML attributes preparation

The script *02_static_data_preparation.py* uses *cml_metadata.csv* file and creates a file containing CML attributes. In order to make it works, the

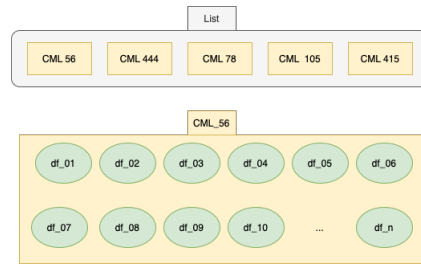


Figure 5.1: Data class output format

there will be error at the end of training. All other variables should have self explanatory names, so every users must adapt them accordingly.

Data. In order to use data we have with the **Training class** we must create a class that inherits from the **DataBase** class in the *data_interface.py* script. In this way we can create any data we want and then use this interface to make the data usable by the Training objects. The DataBase interface loads CML attribute data and matches CMLs with their respected references. Then, it defines four abstract methods that must be implemented by every new data class. The methods are:

- `get_training_data()`
- `get_validation_data()`
- `get_validation_data()`
- `get_cml_attributes()`

Their unified output type is important. The output must be a list containing lists of Pandas Dataframes as shown in 5.1.

If we use continuous data, then each list in the list of lists contains multiple Dataframes each containing continuous data. If we use non-continuous data, then each list in the list of lists contains single Dataframe.

We have two data classes already implemented. The *continuous_data.py*, which operates on continuous data and *all_in_one_dataframe_data.py* which operates on the non-continuous data. If anyone wants to prepare data in any other way, just create a new class, inherit from DataBase and use it with the Training class.

The *all_in_one_dataframe_data.py* uses input data defined in *data_paths.py* under the **ALL_DATA** variable. The data is in form of a single Pandas Dataframe saved as a .csv file. The DataFrame is saved as a Timeseries with columns corresponding to CML names.

The *continuous_data.py* input data is defined under the **DYNAMIC_DATA_FOLDER** variable. The path determines a folder consisting of multiple folders, one folder for each CML named after its name. Inside every CML folder, there are multiple Dataframes stored as a Timeseries in .csv file.

Models. Similar to the data interface there is a **Model** interface in the *model_interface.py* script. Each model inherits from it and then it is usable by the Training class. The Model interface inherits the Pytorch nn.Module and add a couple abstract methods. They are:

- `get_model_type()`
- `create_data_loader()`
- `compute_loss()`
- `get_reference()`
- `get_data_from_batch()`

All above methods are documented in the code. The idea is to achieve low coupling between the Train class and a model. Therefore everything model related is handled by the model itself.

■ Results evaluation and visualization

The last module we created is build for models results performance visualization. It is called the *result_analyzer*. The Training class saves information about the model type we trained. It is either a classification or regression model. This information is saved in *END.txt* file when training is finished. This information is used by the *result_analyzer* to know, which metrics to compute and what graphs to create. All results visualizations presented by us were generated by this module.



Bibliography

- [1] Haldun Akoglu. User’s guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, 18(3):91–93, 2018.
- [2] David Atlas and Carlton W. Ulbrich. Path- and area-integrated rainfall measurement by microwave attenuation in the 1–3 cm band. *Journal of Applied Meteorology and Climatology*, 16(12):1322 – 1331, 1977.
- [3] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview . *Bioinformatics*, 16(5):412–424, 05 2000.
- [4] Blandine Bianchi, Peter Jan Van Leeuwen, Robin Hogan, and Alexis Berne. A variational approach to retrieve rain rate by combining information from rain gauges, radars, and microwave links. *Journal of Hydrometeorology*, 14:1897–1909, 12 2013.
- [5] C. Chwala, F. Keis, and H. Kunstmann. Real-time data acquisition of commercial microwave networks for hydrometeorological applications. *Atmospheric Measurement Techniques*, 9(3):991–999, 2016.
- [6] Christian Chwala, Andreas Gmeiner, Wendong Qiu, Susanne Hipp, David Nienaber, U. Siart, Thomas Eibert, M. Pohl, J. Seltmann, J. Fritz, and Harald Kunstmann. Precipitation observation using microwave backhaul links in the alpine and pre-alpine region of southern germany. *Hydrology and Earth System Sciences*, 16:2647–2661, 08 2012.
- [7] M. Fencl, M. Dohnal, J. Rieckermann, and V. Bareš. Gauge-adjusted rainfall estimates from commercial microwave links. *Hydrology and Earth System Sciences*, 21(1):617–634, 2017.
- [8] Martin Fencl, Pavel Valtr, Milan Kvičera, and Vojtěch Bareš. Quantifying wet antenna attenuation in 38-ghz commercial microwave links of cellular backhaul. *IEEE Geoscience and Remote Sensing Letters*, 16(4):514–518, 2019.
- [9] Hai Victor Habi and Hagit Messer. Recurrent neural network for rain estimation using commercial microwave links. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):3672–3681, 2021.

- [10] H. Leijnse, Remko Uijlenhoet, and J.N.M. Stricker. Microwave link rainfall estimation: Effects of link length and frequency, temporal sampling, power resolution, and wet antenna attenuation. *Advances in Water Resources*, 31:1481–1493, 11 2008.
- [11] Christof Lorenz and Harald Kunstmann. The hydrological cycle in three state-of-the-art reanalyses: Intercomparison and performance analysis. *Journal of Hydrometeorology*, 13(5):1397 – 1420, 2012.
- [12] Hagit Messer, Artem Zinevich, and Pinhas Alpert. Environmental monitoring by wireless communication networks. *Science*, 312:713 – 713, 2006.
- [13] Jonatan Ostrometzky and Hagit Messer. Dynamic determination of the baseline level in microwave links for rain monitoring from minimum attenuation values. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(1):24–33, 2018.
- [14] Aart Overeem, Hidde Leijnse, and Remko Uijlenhoet. Country-wide rainfall maps from cellular communication networks. *Proceedings of the National Academy of Sciences*, 110(8):2741–2745, 2013.
- [15] Jaroslav Pastorek, Martin Fencl, Jörg Rieckermann, and Vojtěch Bareš. Commercial microwave links for urban drainage modelling: The effect of link characteristics and their position on runoff simulations. *Journal of Environmental Management*, 251:109522, 2019.
- [16] J. Polz, C. Chwala, M. Graf, and H. Kunstmann. Rain event detection in commercial microwave link attenuation data using convolutional neural networks. *Atmospheric Measurement Techniques*, 13(7):3835–3853, 2020.
- [17] Marc Schleiss, Jörg Rieckermann, and Alexis Berne. Quantification and modeling of wet-antenna attenuation for commercial microwave links. *IEEE Geoscience and Remote Sensing Letters*, 10(5):1195–1199, 2013.