Low-dimensional Representation Learning for Wireless CSI-based Localisation

Artan Salihu†, Stefan Schwarz†, Aggelos Pikrakis‡ and Markus Rupp†
† Institute of Telecommunications, Technische Universität (TU) Wien
‡Department of Informatics, University of Piraeus, Greece
Email: {artan.salihu,stefan.schwarz,markus.rupp}@tuwien.ac.at, pikrakis@unipi.gr

***Abstract*—In this work, we investigate the potential of deep feedforward neural networks for user position estimation for a multi-path directional channel model in presence, as well as absence of the line of sight path. Furthermore, we take advantage of a *triplet network* architecture combined with a triplet loss function, which allows us to exploit intrinsic properties of channel state information. We show that despite the high-dimensional nature of CSI, the proposed network can be trained to learn from the low-dimensional space and with less amount of training samples compared to the long-established approaches in the literature for fingerprinting localization. Finally, in order to investigate the performance and emphasize the benefits of triplet loss, we compare it to another network based on classification.**

***Index Terms*—Localization, Machine Learning, Deep Learning, Massive MIMO.**

I. Introduction

With an ever increasing demand for high accuracy and precision location based services, wireless positioning of a transmitter is an important research direction for the fifth generation (5G) networks. Location information enables commercially, as well as non-commercially motivated context-aware services. For example, improvements in efficiency for asset and freight tracking in the railway sector highly depend on a position-related information infrastructure [1]. Moreover, location information can be harnessed by the wireless network infrastructure itself, in order to improve availability of the services and adapt allocation of the resources based on a short-term demand [2].

High-quality positioning methods that are based on global navigation satellite systems (GNSS) continue to be subjected to two disadvantages. First, such methods do not provide reliable position estimates when the target is not in line-of-sight with the orbiting satellites. In addition, they require battery-hungry sensors to be active on the target devices. As the trend of communication systems moves towards network densification, propagation conditions will likely become even more challenging and channel characteristics will significantly vary between different locations.

On the other hand, 5G systems will heavily rely on massive multiple-input multiple-output (MIMO) technology, in order to improve spectral efficiency [3]. They operate with a higher density of directionally aware antennas at the base station (BS), which opens up new opportunities for different sensing applications, as well as complement GNSS based methods for position estimation of user equipments (UEs) [4], [5]. Furthermore, due to a large number of antenna elements at the BS, a considerable amount of channel state information (CSI) can be obtained. This information can be harnessed by machine learning (ML) techniques for position estimation of the transmitter.

Unlike long-established localization methods that rely on the received signal strength indicator (RSSI), CSI is able to distinguish multipath characteristics, and thus has encouraged active research towards more robust and higher accuracy localisation schemes [6]. Obtained high-dimensional CSI at a large-antenna BS provides finer granularity of UE signatures and can reveal spatiotemporal information about the transmitter itself, as well as its surrounding.

Several approaches that make use of CSI fingerprinting with machine learning and, in particular deep learning, have been proposed recently [7]–[11]. All of these methods utilize a neural network to learn a function that maps the input CSI to a position. Albeit these approaches report to achieve high accuracy, they also heavily rely on a large amount of samples collected at high-spatial resolution. Acquisition of such a large database of CSI fingerprints can be reasoned by the practical potential of the crowdsourcing paradigm [12]. However, taking into consideration the temporal aspects of propagation conditions, the amount of available training data for (re-)training the network at smaller time intervals is a limiting factor for practical implementations.

Motivated by the success of few-shots learning in different applications such as vision recognition [13], object detection [14], object similarity [15], to name a few, we propose and investigate a similar approach to a so called *triplet network*, which is inspired by Siamese networks [16]. A Siamese network has been recently proposed in wireless positioning for *channel charting* technique [17] too. However, the goal of [17] is to maintain the distance between transmitters by minimizing the squared distance error. In contrast, we strive to learn the low-dimensional representation of CSI in a way that we employ a *triplet loss* objective function, which has shown outstanding performance in face recognition and clustering [18]. In Section II, the proposed architecture is described in detail so that recognizing the differences from traditional designs are easy to perceive.

In order to investigate the performance and emphasize the benefits of a triplet loss, we develop another network that is based on a conventional formulation of the problem, i.e., a

classification task. We refer to this as the *base network*.

We structure the remaining of the paper as follows. In Section II, we introduce the system model considered for this work. In Section III, we describe the details on the proposed network architecture and localization methods. In Section IV, we demonstrate the potential of the triplet loss for wireless CSI based localisation by performing simulations for different propagation conditions. Finally, in Section V we draw our conclusions.

II. System Model

We consider N single-antenna transmitters corresponding to $\{\mathbf{x}_n \in \mathbb{R}^2\}_{n=1}^{N}$ locations; i.e., position coordinates of the users. We assume that the user locations are distributed based on a Poisson point process (PPP) with the density λ_n. Hence, the probability of having N users in the plane is $\mathbb{P}(N, \mathcal{A}) = [\lambda S(\mathcal{A})]^n e^{-\lambda S(\mathcal{A})}/N!$ [19], where $S(\mathcal{A})$ is the area of the bounded region of interest $\mathcal{A}$ (ROI). We consider a single BS at a known position with N_{BS} antenna elements.

The signal from each UE that is received at N_{BS} antenna elements contains N_{SC} subcarriers. We assume that in addition to a direct line-of-sight (LOS) link between the BS and UE, there are P additional NLOS paths due to possible scatterers or reflectors in the ROI. We assume that UEs, as well as scatterers are static for the duration of a transmission. For a single coherence bandwidth and $P+1$ paths, the $N_{\mathrm{BS}} \times 1$ multi-path channel vector for the subcarrier n is modelled as [20]

$$\mathbf{h}[n] = \mathbf{A}_{\mathrm{BS}}[n]\Gamma[n], \text{ where} \tag{1}$$

$$\mathbf{A}_{\mathrm{BS}}[n] = [\mathbf{a}_{\mathrm{BS}}(\varphi_0), \ldots, \mathbf{a}_{\mathrm{BS}}(\varphi_P)] \tag{2}$$

is the BS steering vector matrix, which introduces the angle of arrivals at the receiver φ_ℓ for each path $\ell = 0, \ldots, N_P$. If the LOS exists, in addition to the NLOS paths, $\ell = 0$, we will refer to this as NLOS case. In contrast, if the received signal is obtained only by the NLOS paths, i.e., $\ell = 1, \ldots, P$, we will refer to this as the only non-line of sight (ONLOS) case. In addition,

$$\Gamma[n] = \sqrt{N_{\mathrm{BS}}} \times \begin{bmatrix} \alpha_0 \mathrm{e}^{\frac{-j2\pi n \tau_0}{N_{\mathrm{SC}} T_s}} \\ \vdots \\ \alpha_P \mathrm{e}^{\frac{-j2\pi n \tau_P}{N_{\mathrm{SC}} T_s}} \end{bmatrix}, \tag{3}$$

where $\alpha_p = h_p/\sqrt{\rho_p}$, with h_p, ρ_p and τ_p denoting the random complex channel gain, path loss and the time of flight (TOF) for the pth path, respectively. Finally, in this work we consider a uniform linear array antenna (ULA) geometry with $d_{an} = \lambda_c/2$ equidistant elements

$$\mathbf{a}_{\mathrm{BS}}(\varphi) = \frac{1}{\sqrt{N_{\mathrm{BS}}}} \left[1 \; e^{j\frac{2\pi}{\lambda_c} d_{\mathrm{an}} \sin\varphi} \cdots e^{j(N_{\mathrm{BS}}-1)\frac{2\pi}{\lambda_c} d_{\mathrm{an}} \sin\varphi}\right]^T. \tag{4}$$

A. Reference Scenarios

In contrast to N unknown user locations and N_P scatterers, we assume that R locations $\{\mathbf{x}_r^{(RP)} \in \mathbb{R}^2\}_{r=1}^{R}$ are known in advance. We will be referring to these locations as the reference point locations (RPs). We will utilize the coordinates of these RPs for our training purpose. Moreover, we consider that RPs are distributed with certain density, λ_r, in the area such that $S(\mathcal{A})$ is partitioned into:

1) R squares. Each square denotes a quantized location and determines the resolution, accuracy, as well as the overall computational complexity of the system. Throughout this paper, we will refer to this scenario as the *grid scenario*.
2) While the *grid scenario* allows us to control the number of samples per each location for a more accurate investigation of the performance of our schemes, it is obviously impractical. In contrast to the first scenario, we consider that RPs are distributed based on PPP and the $S(\mathcal{A})$ is a Voronoi tessellation. We refer to this scenario as the *Voronoi scenario*.

Collecting a large number of CSI from R squares or cells might be considered as challenging and time-consuming. However, with the widespread use of mobile devices, traditional site surveying of the area, i.e., using specialized equipment and experienced operators, can be replaced by crowdsourcing approaches. In this case, users can participate to share their accurate locations. Nevertheless, developing a system with minimal requirements for ground-truth information for training improves its overall availability and reliability. Next, we will describe the proposed methods for localization using neural networks.

III. Triplet Network and Localization

In this section, we describe the model, techniques and localisation algorithm to estimate the position of the unknwon transmitter. First, we will describe the details of deriving the *base network*, which is depicted in Fig. 1a. Furthermore, before feeding CSI into the network, we explicitly calculate the average per antenna over all subcarriers. In addition, we handle CSI complex values as two independent real numbers. Thus, our input channel vector is $\hat{\mathbf{h}} \in \mathbb{R}^M$, where $M = 1 \times 2N_{\mathrm{BS}}$ is the original dimensionality.

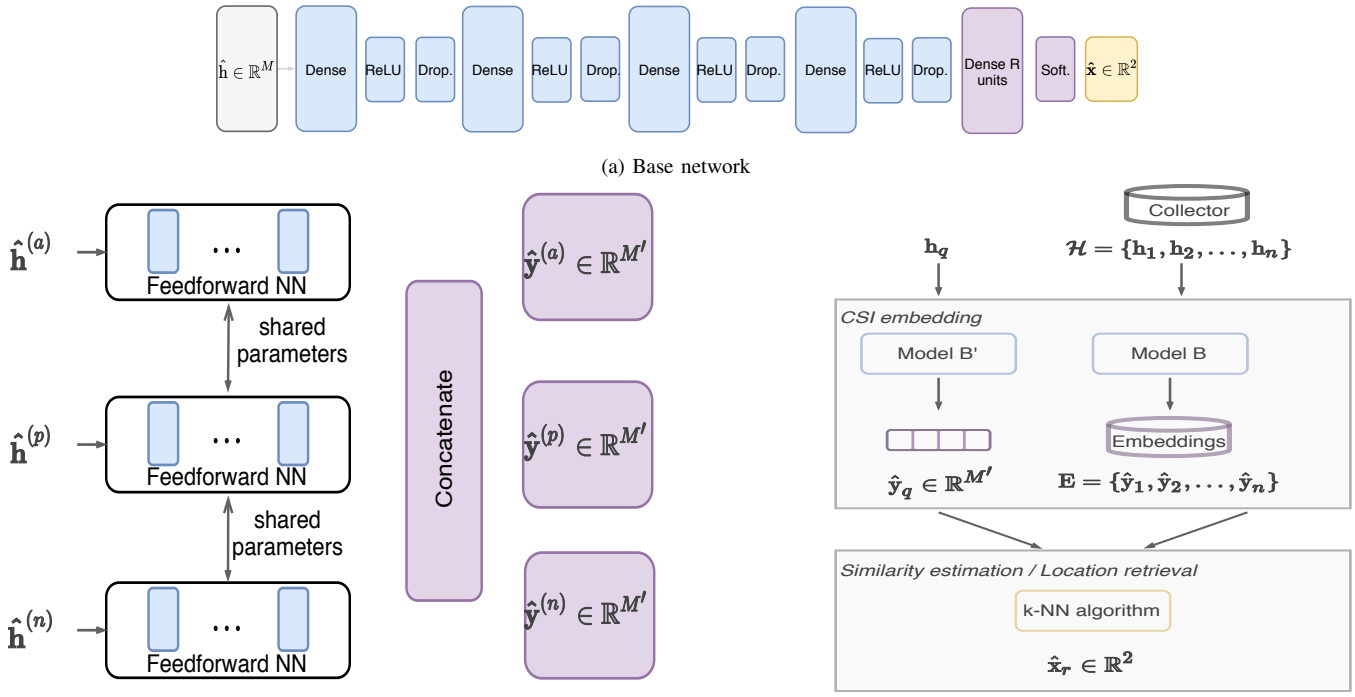
A. Base network

We use a deep feedforward network, which is a commonly used structure for neural networks. The structure of the network has an input layer that corresponds to the CSI values that are being evaluated, a number of intermediate computation layers and finally an output layer. So, the goal of such a network is to approximate some function that maps an input to a category or another output type

$$f\left(\hat{\mathbf{h}}; \boldsymbol{\theta}\right) : \mathbb{R}^M \mapsto \mathbb{R}^{M_L}. \tag{5}$$

Each layer $\ell = 1, \ldots, L$ has the responsibility of mapping from the previous to the next layer. The mapping depends on the output of the previous layer, $\mathbf{z}_\ell = f_\ell(\mathbf{z}_{\ell-1}; \theta_\ell)$, and on a set of parameters $\theta_\ell = \{\mathbf{W}_\ell, \mathbf{b}_\ell\}$, where $\mathbf{W}_\ell$ and $\mathbf{b}_\ell$ are the weights and biases respectively. Furthermore, the $\ell = 1$ layer maps the values from the input layer, $\mathbf{z}_0 = \hat{\mathbf{h}}$.

Computations and mapping in each layer are done via nodes (i.e., neurons), where each node is connected to every other



(a) Base network

(b) Triplet network

(c) System overview based on triplet network

Fig. 1: Proposed method using *base network* and *triplet network* for user position estimate

node of the neighbouring layers. In addition to summing up the weighted inputs, each neuron also applies a non-linear activation function, $f_\ell\left(\mathbf{z}_{\ell-1};\theta_\ell\right) = \sigma\left(\mathbf{W}_\ell \mathbf{z}_{\ell-1} + \mathbf{b}_\ell\right)$. The modelling potential of a neural network also depends on the choice of the activation function, $\sigma(\cdot)$.

Training a neural network describes the goal of finding the set of optimal parameter values that minimize a given loss function, $\mathcal{L}$, over a given batch of dataset, S_t, and its corresponding desired output,$\mathbf{y}$,

$$J(\boldsymbol{\theta}) = \frac{1}{S_t}\sum_{i\in S_t}\mathcal{L}\left(\mathbf{z}_{L,i}, \mathbf{y}_{L,i}\right), \tag{6}$$

where $\mathbf{z}_{L,i}$ is the output of the last layer when $\mathbf{z}_{0,i}$ is used in the input layer. The set of parameters can be efficiently computed with the stochastic gradient descent algorithm (SGD) [21] implemented in many software libraries. The SGD employs backpropagation, which calculates the loss between the predicted output and the actual output, and updates the parameters iteratively

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta\nabla J\left(\boldsymbol{\theta}_{t-1}\right) \tag{7}$$

using an adaptive or a static learning step size, η. We used Adam [22] implemented in Tensorflow [23] with static learning rate, $\eta = 0.001$.

The *base network* has $L = 5$ layers. To avoid the issue with the vanishing gradient [24], $\sigma\left(z_i\right) = \max\left(0, z_i\right)$, is adopted as the non-linear activation function for the layers $l = \{1, 2, 3, 4\}$. For the output layer, we use a softmax activation function, which maps the real-valued input into prediction probability scores in the range of $[0, 1]$

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{r=1}^{R} e^{z_r}}, \tag{8}$$

where each score is the desired prediction that corresponds to the correct RP location. The number of nodes for each intermediate layer is 650. Whereas the number of nodes for the output layer is equivalent to the number of RP locations, R. As we have not many hyperparameters, we utilize a grid search [21] to choose values that yields best results. We employ cross-entropy as loss function to train this classification network. In order to avoid overfitting, we also adopt a squared Frobenious-norm regularization on the weights for intermediate computation layers with a certain regularization strength, λ_S,

$$\mathcal{L} = -\sum_{r=1}^{R} y_r^{(RP)}\log(\hat{y}_r) + \lambda_S \sum_{\ell=1}^{4}\left\|\mathbf{W}^{(\ell)}\right\|_F^2, \tag{9}$$

where $y_r^{(RP)}$ and $\hat{y}_r$ are the desired and the predicted output scores, respectively.

Furthermore, we employ a dropout between the intermediate computation layers [25]. In our case, we use $p = 0.8$, where p denotes the fraction of active node connections.

Finally, for the purpose of localization, the system is divided into two phases: *offline* and the *operation* phase. During the *offline* phase, the network is trained using the collected CSI samples at each R location. Then, the estimated position of the unknown transmitter during the *operation* phase, $\hat{x}_i$, is a

weighted mean over the K highest predicted probability scores that correspond to known reference point locations

$$\hat{\mathbf{x}}_i = \frac{\sum_{k=1}^{K} \hat{y}_{i,k} \mathbf{x}_{i,k}^{(RP)}}{\sum_{k=1}^{K} \hat{y}_{i,k}}. \tag{10}$$

The value of K depends on the overall classification accuracy of the network. We consider $K = 2$ as most of the probability scores will be zero and thus we can avoid computational complexity by considering all the scores for the weighted mean.

B. Triplet network

In contrast to the conventional formulation of the problem, i.e., classification task for the *base network*, we propose a new approach that is based on learning low-dimensional representation of the CSI using the *triplet network* as depicted in Fig. 1b. The network is composed of three branches of deep feedforward networks. Each of these networks uses the same hyperparameters as the *base network* for the intermediate computation layers. For the output layer, we use the identity function, $\sigma(z_i) = z_i$. Moreover, all the three branches share the same parameters. Thus, the computation complexity, in terms of the number of parameters, does not increase with increasing number of RPs. This is another advantage compared to other solutions that define the problem as a classification task. The goal of the *triplet network* is to learn an embedding in the way that the squared distance between entities that are "related" to each other is less than the square distances of those that are "unrelated" [18]. The term "related" in our case is interpreted with respect to the CSI that are obtained from transmitters from the same R. Whereas "unrelated" corresponds to all pairs from other locations. Therefore, the objective function encourages that a CSI embedding obtained from a location within R to be more *similar* to another embedding of the same location rather than that from any other one.

To train the network, we sample similar and non-similar CSI, i.e., if two CSI vectors correspond to the same *square* or *cell* they are considered similar, otherwise they are not similar. Thus, we need to sample triplets in such a way that for every CSI inside the RP region, we choose one CSI feature vector as the anchor node and the other one as the positive sample. Then, for the negative (not similar CSI), we only need to randomly select a feature vector that does not belong to the same quantized location as the anchor node. For large scale scenarios and/or very high density of RPs, it might not be computationally efficient to sample all the possible triplets. Therefore, one could sample only a portion of the triplets for the sake of the training process.

By collecting all the possible triplets from the ROI under investigation, we minimize the loss function:

$$\mathcal{L} = \max\left(0, \left\|\hat{\mathbf{y}}^{(a)} - \hat{\mathbf{y}}^{(p)}\right\|_2^2 - \left\|\hat{\mathbf{y}}^{(a)} - \hat{\mathbf{y}}^{(n)}\right\|_2^2 + \delta\right) \tag{11}$$

where $\hat{\mathbf{y}}^{(a)}$, $\hat{\mathbf{y}}^{(p)}$ and, $\hat{\mathbf{y}}^{(n)}$ denote the low-dimensional representations of CSI for anchor node, positive sample and negative sample, respectively. The dimensionality of respective embeddings is $M' < M$. For the purpose of this study, we have chosen $M' = 4$. The hyper parameter δ enforces a margin between the similar and non-similar pairs. This parameter is tuned during the training process and for most of the experiments $\delta = 0.2$ yields the best results.

Similar to the first method, we divide the system into *offline* and *operation* phase. During the *offline* phase, the *triplet network* is trained as described above. However, in contrast to the first approach, during this phase we need to save the derived embeddings in addition to the coordinates of the RPs in an embedding collector. This can be observed as additional overhead for the storage and computation parts of the system. However, due to the capability of the network to be trained using low-dimensional representations of the CSI, M', this can be considered feasible for many modern systems, as well as applications. During the *operation* phase, only one branch of the network is required to estimate the embedding for the CSI generated by an unknown transmitter. In this case, given that a new UE appears in the ROI, we form the distance metric for each RP as:

$$d_r = \| \hat{\mathbf{y}}_q - \hat{\mathbf{y}}_r \|, \tag{12}$$

where $\hat{\mathbf{y}}_q$ is the predicted embedding CSI of new UE. Then, we choose $r_{NN} = \arg\min_r d_r$. Finally, the location of the RP with minimum distance is selected as the position of the transmitter, $\hat{\mathbf{x}}_q = \mathbf{x}_{r_{NN}}^{(RP)}$. The process of training, as well as operation is depicted in the Fig. 1c.

IV. Simulation Results

In this section, we evaluate and compare the performance of the aforementioned localisation algorithms due to the various factors that influence their ability for position estimate of the unknown transmitter. First, the performance is evaluated for the positioning task for two different propagation cases, i.e., when the line of sight path is present and absent, as well as with respect to the impact of density of reference point locations. Second, the influence of noisy measurements and the number of antenna elements is further investigated. Afterwards, the ability of the network to adapt in different distribution scenarios of the RPs is studied. Finally, the amount of samples required for training the *triplet network* in contrast to *base network* is explored.

The metric used to evaluate the overall performance of predicted position for the unknown transmitters is the root mean squared error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{S_{\mathrm{test}}} \sum_{n=1}^{S_{\mathrm{test}}} \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|^2}, \tag{13}$$

where S_{test} is the test dataset size, $\hat{\mathbf{x}}_n$ and $\mathbf{x}_n$ are the coordinates for predicted and actual position of the transmitter respectively. For the simulations, the model with the lowest validation loss from 100 epochs is selected. Furthermore, when LoS is absent, the triplet loss margin during training is changed to $\delta = 10$. The dataset is split into 0.8 and 0.2 for training and

testing respectively. The scenario considered for simulations is $40m \times 40m$ area, the BS at the origin $(0, 0)$, and user positions distributed in the far field at minimum distance of $20m$ from the BS. Number of non-line of sight paths is $P = 3$, and the number of subcarriers is $N_{\mathrm{SC}} = 512$.

A. Line of sight path and density of RPs

The density of reference point locations is the parameter that determines the desired positioning accuracy. With the increase of RPs, the ability of the system for higher accuracy increases too. However, the presence or the absence of line of sight path in a multi-path channel influences the accuracy of the prediction and consequently limits the accuracy even when high density of RPs is used.

In Fig. 2, we show the impact of line of sight with the increase of density of RPs for both networks. When the line of sight path is present, the increase in density of RPs increases the overall prediction performance. Furthermore, the *triplet network* outperforms the *base network* when the density of RPs is $\lambda_r > 0.05$. However, when the line of sight path is absent, the overall accuracy drops for higher density of RPs compared to the case when line of sight path is present.

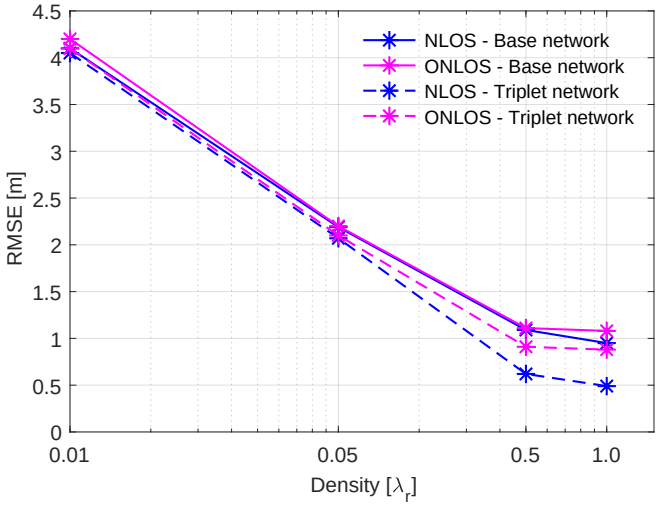


Fig. 2: The influence of line of sight path and the density of reference point locations for *base network* and *triplet network*.

B. Influence of noisy CSI and the number of antennas

The evaluation of the performance under noisy CSI measurements and different number of antenna elements for the *triplet network* is shown in Fig. 3. We can see that the methods can provide acceptable accuracy for relatively low SNR. Furthermore, the number of antenna elements in the BS noticeably influences the position accuracy [26], which is also observed in our simulation results. However, the advantage of using higher density of antenna elements could become even more evident in case of sampling at higher spatial resolution.

C. Distribution of reference point locations

As we discussed in the Section II, collecting ground truth under the assumption of a *grid scenario*, where each square is a quantized location and corresponds to one RP, is not practical. Thus, in Fig. 4 we show that the networks, without

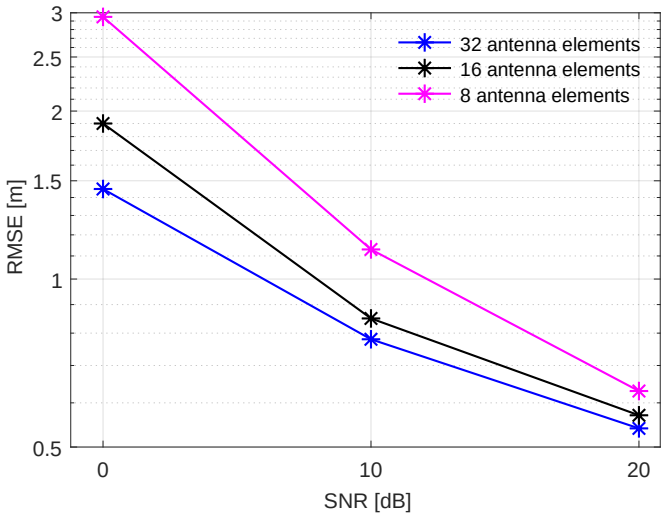


Fig. 3: The impact of noisy CSI measurements and the number of antenna elements.

any hyperparameter changes, can be trained and provide similar performance when a *Voronoi scenario* is taken into consideration for investigation of accuracy. We can observe that when the line of sight path is present and low density of RPs, both models can provide an error of less than $10\mathrm{m}$ for *grid*, as well as *Voronoi scenario*. In contrast, increasing the density of RPs to $\lambda_r = 1.0$, the *triplet network* outperforms the *base network* for both types of scenarios.

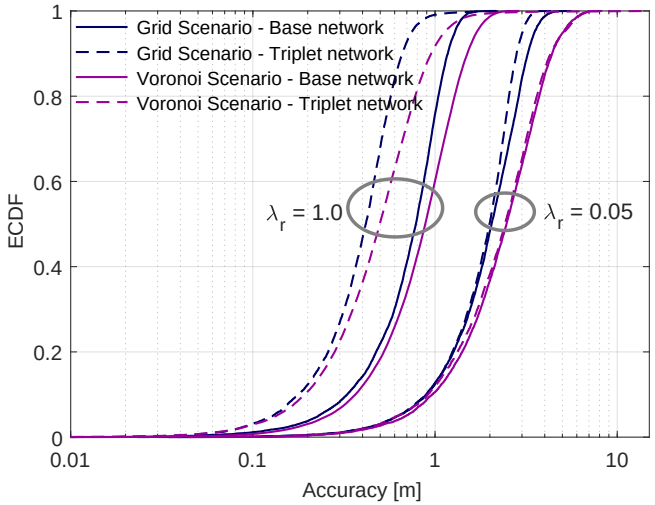


Fig. 4: *Triplet network* outperforms *base network* for a high density of RPs in *grid* and *Voronoi scenarios*.

D. Amount of training samples

Developing a machine learning algorithm that allows us to train with as few groundtruth labels as possible is an important factor for maintaining high availability. Obtained CSI from the reference point locations in a dynamic scenario can become outdated quickly. This requires for the ability of the network to be trained at smaller times scales depending temporal characteristics of the environment.

In Fig. 5 we show the investigation results for the amount of samples required for (re-)training the networks. As we can

observe, the *triplet network* can provide satisfactory positioning accuracy even with very low amount of samples, i.e., 3 samples /m^2 corresponding to a reference location. The *triplet network* outperforms the *base network* at the $80\mathrm{th}$ percentile even when the after-mentioned network is trained with almost 10 times the amount of samples. This is a remarkable potential of these types of networks towards a high-accuracy practical localisation scheme in an ever-changing environment.

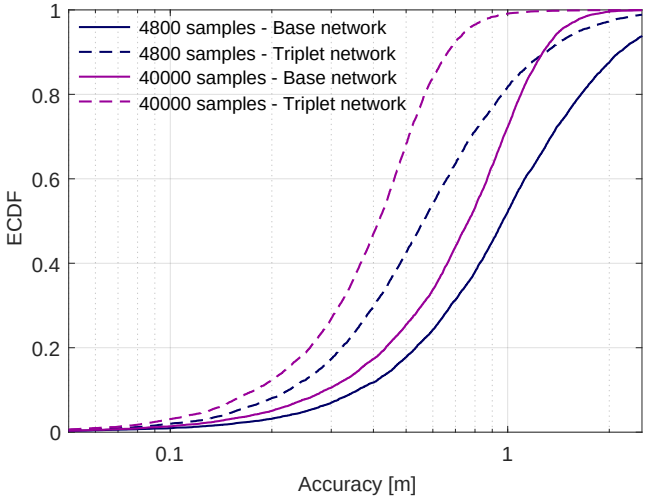


Fig. 5: Accuracy and the amount of samples for training a *triplet network* and a *base network*.

V. Conclusion

In this paper, we investigated the potential of feedforward neural networks in a *triplet network* architecture towards a high-accuracy localisation scheme depending on the density of reference point locations. We showed that the proposed scheme can provide higher accuracy with less amount of samples for (re-)training, enabling the potential of neural network based localisation for practical implementations in a changing propagation environment. We also investigated the impact of the presence and the absence of line of a sight path in a multi-path channel model. We showed that we can learn a low-dimensional representation of CSI acquired at a massive MIMO base station, which is sufficient to capture multi-path characteristics of the channel and still reveal spatial signature of the unknown transmitter.

References

[1] 3GPP, "Study on positioning use cases," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 22.872, 04 2018, version 16.1.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3280

[2] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, 2014.

[3] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—what is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, 2019.

[4] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2475–2487, 2017.

[5] M. Rupp and S. Schwarz, "An LS localisation method for massive MIMO transmission systems," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4375–4379.

[6] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 25, 2013.

[7] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.

[8] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, "Deep learning-based indoor localization using received signal strength and channel state information," *IEEE Access*, vol. 7, pp. 33 256–33 267, 2019.

[9] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2016.

[10] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18 066–18 074, 2017.

[11] M. Widmaier, M. Arnold, S. Dorner, S. Cammerer, and S. ten Brink, "Towards practical indoor positioning based on massive mimo systems," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–6.

[12] W. Zhao, S. Han, R. Q. Hu, W. Meng, and Z. Jia, "Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3236–3247, 2018.

[13] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[14] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[15] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[16] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[17] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 200–207.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[19] D. Moltchanov, "Distance distributions in random networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1146–1166, 2012.

[20] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave mimo systems," *IEEE journal of selected topics in signal processing*, vol. 10, no. 3, pp. 436–453, 2016.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[24] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] L. Gui, M. Yang, H. Yu, J. Li, F. Shu, and F. Xiao, "A cramer–rao lower bound of csi-based indoor localization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2814–2818, 2017.